TATA CONSULTANCY SERVICES

Java Lounge

Logout

Java

Course Completion Quiz

1.IT Application Overview And Features

Objective

- Introduction to Application Understanding
- Different types of Applications with Examples
- Basic Application Architecture

1.4 Basic Application Functions - CRUD

1.5 Reference Links & videos

1.1. Introduction to Application Understanding

If you manage your work without any mess with the help of alarms in a smart phone

Figure 1

Or if you are reading any posts with your tablet,

Figure 2

You are using Applications/App.

So what exactly does an Application mean?

An application is a type of software that allows you to perform specific tasks which otherwise is very difficult to perform manually.

Before we go any further think about few computer applications you are familiar with.

Sticky Notes

You can use Sticky Notes to write a to-do list, jot down a phone number, or to do anything else that you would you use a pad of paper for.

Figure 3

Windows Notepad

Notepad is a basic text editor that you can use to create simple documents.

Figure 4

Whatsapp

WhatsApp Messenger is a proprietary, cross-platform instant messaging subscription service for smart phones.

Figure 5

Facebook Application

A social utility that connects people with friends and others who work, study and live around them

Figure 6

IRCTC Portal

Online Passenger Reservation System provides booking facility of Railway tickets online

Figure 7

By giving a closer look to these applications you will find that few of the applications are specially designed for Desktop/Personal computer while few others only work on mobile devices. Similarly you will find that in order to run certain other applications you need a web browser like Internet Explorer, Chrome etc.

Ask a doubt *(Misuse of 'Ask a Doubt' Section will be dealt as per the Terms & Conditions of Campus Commune)*

Open Doubts | **Closed Doubts**

PRIYANKA

I have just submitted my quiz and on the last question ie. 20th question it was showing 19 out of 20 and when i have submitted the quiz my score was 70 instead of 78. I cannot understand what actually has happened. Suggestions are highly helpful.

about 2 months ago

1.2. Different types of Applications with Examples

Broadly applications are divided broadly into three types

i. Desktop Applications
ii. Mobile Applications
iii. Web Applications

Desktop Applications – Desktop Applications are applications that runs stand alone in a desktop or laptop computer.

Eg. Microsoft Paint

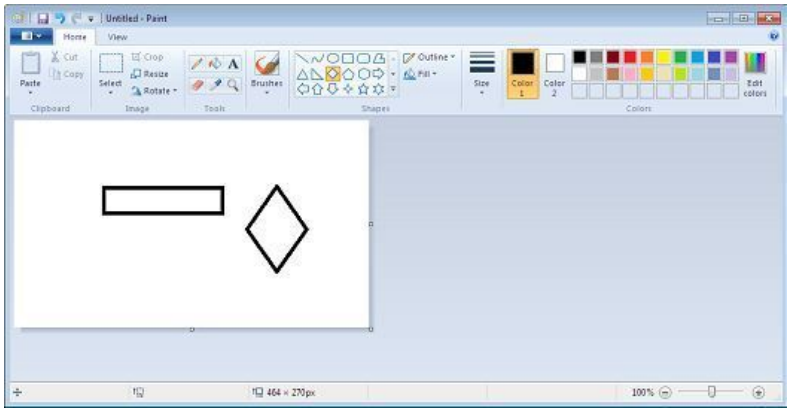


Figure 8

Figure 9

Mobile Applications

Desktop and laptop computers aren't the only devices that can run applications. You can also download applications for mobile devices like smart phones and tablet computers, which opens up a lot of new possibilities. Here are a few examples of mobile applications:

Google Wallet : is a mobile payment application

Figure 10

ICICI Mobile Banking Application

Figure 11

Web Applications - are applications that are accessed over a network such as the Internet or an Intranet.

Eg : ICICI Net Banking application

Figure 12

Passport Seva Portal to deliver passport services to citizens in a timely transparent manner.

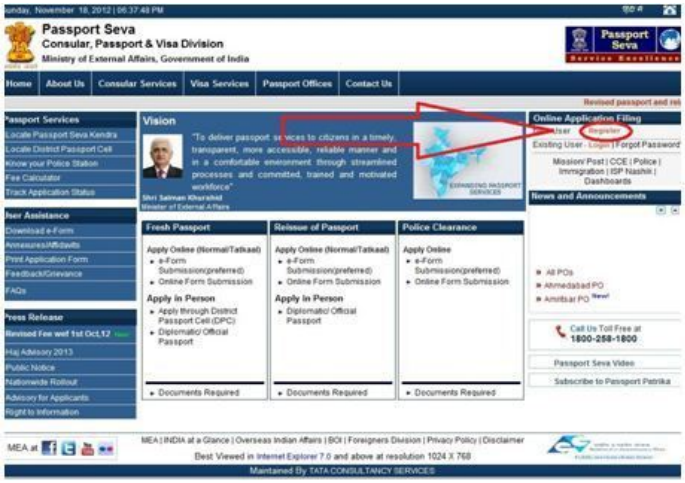


Figure 13

Ask a doubt *(Misuse of 'Ask a Doubt' Section will be dealt as per the Terms & Conditions of Campus Commune)*

Open Doubts | **Closed Doubts**

SUMANTA SAHA

According to above study material, "Web applications are applications that are accessed over a network such as the Internet or an Intranet". Does that make every website a web application?

about 23 days ago

HARIPRASAD R

Not necessarily. Only a site that performs a particular work. A simple static site cant be taken as an application,

ABDUL HAQ

Some of the apps on mobile run based on internet such as Whatsapp, Facebook messanger etc.,. Are such apps called as Mobile Applications or Web Applications?

about 2 months ago

LAKSHMINADHA PRASAD THOTA

If any application which is installed in the mobile and that should be run with out depended on internet or any other external sources is called mobile application.

example:- if we install any mobile game,or dictionary that will run individually with out depending on any web server or internet.

Whatsapp and facebook are depending on internet and some other servers so they called web applications.

LAKSHMINADHA PRASAD THOTA

Web Applications - are applications that are accessed over a network such as the Internet or an Intranet.

LAKSHMINADHA PRASAD THOTA

Small correction to my above replay..

Whatsapp and facebook both are installed in mobile application..so any application that is installed on mobile consider to be mobile application only....because their user interface is acced through mobile only....

KALYAN PILLA

Web applications are applications with a web interface. Think of it this way- *Thunderbird* is a **Desktop application**, gmail(*web-mail*) that you access through a browser is a **Web application** and the *gmail app* on your phone is a **Mobile application**. All the three perform the same task of delivering your mail from a remote server to you.

The above classification is NOT based on what an app does or how it works, rather on the platform on which an application works.

ABDUL HAQ

What is the difference between internet and intranet? If there is any difference between them, please elaborate them.

about 2 months ago

APURV KUMBHARE

http://compnetworking.about.com/cs/intranets/g/bldef_intranet.htr

Refer this link..

VEERABHADRASWAMY BODA

internet world wide collection of inter collected network whereas intranet is group of computers connected to use internet . Both use TCP/IP protocol . But the range varies intranet is limited to those group of computers . Intranet provides security .

ALURI REDDY

Internet is a broad public network that any one can access through an internet service provider while in contrast, an intranet is a private network maintained by an organization for the use of it's own employee's,members or others who are explicitly permitted access.

ABDUL HAQ

Can any one please tell me, to which application does facebook belongs to????????

about 2 months ago

VINDHYA BODDUPALLI

facebook is a web application

SAI YERIGERI

facebook messenger and whatsapp are mobile based applications, because they work only on mobile.

SREE SETTURU

facebook is a mobile application. nw we have a seperate facebook app like whatsapp

UDAYADITYA BORUAH

Facebook, in the general falls into both the categories of mobile as well as web applications. If it is used as a

mobile app on the smartphones,it is a mobile application. However when accessed with the desktop or laptop it is a web application.

LAKSHMINADHA PRASAD THOTA

Its again depends on context and the way you access the face book..

if you access the facebook using web browser by giving facebook url that comes under web application.

if you install facebook app in your mobile and access the facebook that comes under mobile app.

KALYAN PILLA

Facebook is an online service that allows you to socialize and connect with people. You could access this service using

- A **Mobile application** (facebook app)
- The **Web application** (via https://www.facebook.com or https://m.facebook.com on you desktop, laptop, mobile or any other Web connected Device)
- Some **Desktop application** (eg. facebook XMPP on pidgin)

ROHAN DUTTA

FACEBOOK is a **web application** because we are using an web browser where we have to provide an url. But if you are using **FACEBOOK MESSENGER** or **FACEBOOK APP** from your smartphone or tablet then it is a **mobile application.**

JITENDRA CHOUDHARY

As MOBILE APP : If one has installed Facebook App in his mobile phone/handset/tablets and accessing the facebook from there, then it is an MOBILE APP. Mobile app is developed in the native programming languages like Java, C++. Also these days, mobile apps can be developed with HTML5 ans its sets of API provided by the mobile OS. Mobile apps use REST or SOAP APIs to communicate with the server.

As Web App : When we are accessing the facebook from web browsers like Internet Explorer, Google Chrome, then it is an web app. We are typing in broser's address bar http://www.facebook.com to access and use it. In this case at its front-end, UI(user-interface) is built using HTML,CSS, and JavaScript.

In mobile app case, we will not see the domain facebook.com directly anywhere, we directly interact with app after login, all domain bases internet access is being done by the mobile app internally. All facebook does, is providing data in each version of app.

TATA CONSULTANCY SERVICES

Java Lounge

Logout

Java

1.IT Application Overview And Features	Q
2.Need of Programming and Introduction to OOP approach	Q
3.Introduction to Java,JVM,JDK	Q
4.Operators and Variables in Java	Q
5.Introduction to class in java,access specifiers, this and static in Java	Q
6.Java Library, Packages, Use of import	Q
7.Conditional operations	Q
8.Basic Iterations and Arrays	Q
9.Designing Web Pages Part 1(HTML and CSS)	Q
10.Designing Web Pages Part 2 (JavaScript and JQuery)	Q

Course Completion Quiz

1.3. Basic Application Architecture

At the highest and most abstract level, the logical architecture view of any system can be considered as a set of cooperating components grouped into layers as given in the below figure.

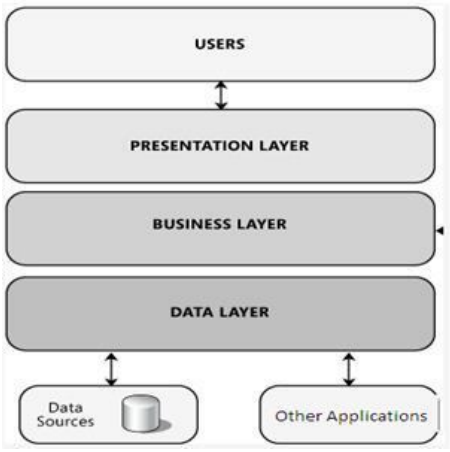


Figure 14

Presentation Layer / User Interface layer

The top most layer of the application through which end-user interacts with the system. The main function of this layer is to translate the tasks and results to something which the end user can understand.

Business layer

This layer implements the core functionality of the system, and encapsulates the relevant business logic.

Data/Resource layer

This layer stores and retrieves information from the storage systems (databases/file systems). The information is then passed to the Business layer for processing and then eventually back to the user.

Layers and Tiers of an Application

Layer

Irrespective of the type of application that you are designing, and whether it has a user interface or it is a services application that only exposes services (not to be confused with services layer of an application), you can decompose the design into logical groupings of software components. These logical groupings are called layers. Layers help to differentiate between the different kinds of tasks performed by the components, making it easier to create a design that supports re-usability of components.

Benefits of layers:

- Simplicity: easy to design once layers and their interaction are defined clearly
- Flexibility: easy to modify and develop networks by separate layers modifications
- Incremental changes: add new layers, add new functions to a layer

Tier

When the logical groups of components are physically separated in different machines, it is known as Tier. The communication between the different tiers of an application is through network.

Single/One Tier Architecture – All layers (Presentation, Business and Resource) are at same place. The application does not require network connection to work. It is easy to install in a system and use it.

Figure 15

Eg: MS Access, MS Excel

2- Tier Architecture - In 2 tier architecture there can be 2 possibilities

1. Presentation Layer in one machine and Business & Resource Layer is available in different machine
2. Presentation & Business layer in one machine and Resource layer in another machine.

Below figure shows a big picture of 2 tier Architecture.

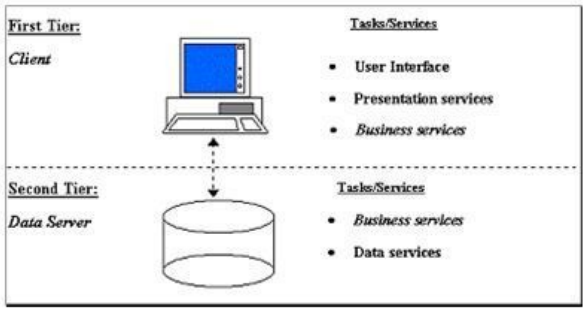


Figure 16

3 – Tier Architecture – In this all three layers of an application are physically separated in different machines.
Eg : Most of the web applications which involves data storage are 3 -tier architecture applications.

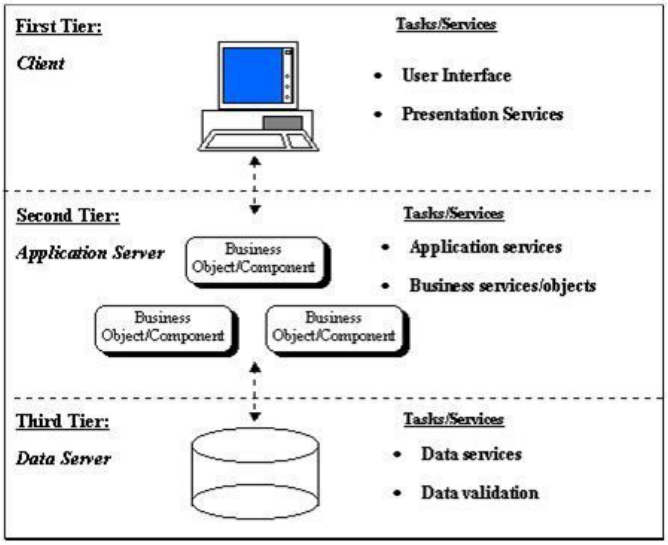


Figure 17

The below figure will help you in visualizing the communication in a web application.
In this example a 3-tier web application system is represented where all three layers of the application is distributed in three different machines.

The Presentation layer is at the client tier and from there request is sent to the Business layer and after processing the request its again sent to Storage/Resource tier for storing.

Figure 18

Multi-tier/ N-tier Architecture: All the 3 -tiered applications can be again physically separated by breaking it into different tiers and establishing a communication between all those tiers.

The terms tier and layer are often used interchangeably. When I use tier, I am referencing a physical layer. When I use layer, I am referencing a distinct software layer. Layers must exist on a tier, and thus tiers contain layers.

Ask a doubt *(Misuse of 'Ask a Doubt' Section will be dealt as per the Terms & Conditions of Campus Commune)*

Open Doubts | **Closed Doubts**

SUMANTA SAHA

The above study material states, "Layers must exist on a tier". Is it possible for a tier to contain more than one layer?

about 23 days ago

VELANTISH V. **Moderator**

Layers are a logical separation and **tiers** are a physical separation. Which means in a layers architecture the presentation, business logic and data layer are on the same machine where as in tiers they are all located on different physical machines.

HEMANT MAHANT

What are the examples of system having 2-tier architecture.?

about 25 days ago

ALURI REDDY

The UNIX print spooler is an example of a two-tier client-server architecture.

The client reads a file to be printed and passes the file's contents to the server. The server performs a service by printing the file.

PALAK PARHI

logical grouping of components? what does components actually refer to?

about 2 months ago

GAMPA SOWMYA

Component actually refers to a reusable part of the software.

ABHISHEK SAH

As stated these are "software" components which are blocks of of instructions(code) each for a different purpose. Similarly their are hardware components such as mouse,CPU , Display etc.

ANUJ PODDAR

Elements that **encapsulate processing and data** in a system's architecture are referred to as software components.
A software component is an architectural entity that

- encapsulates a subset of the system's functionality and/or data
- restricts access to that subset via an explicitly defined interface
- has explicitly defined dependencies on its required execution context

Components typically provide application-specific services.
Put another way, a software component is a locus of computation and state in a system.

1.4. Basic Application Functions - CRUD

Each application gives the CRUD feature in its own way. CRUD is the acronym for Create, Read, Update and Delete.

Create – Adding/Storing any kind of data to the application

Read - There should be a feature where by the data added can be read

Update – There should be a facility to update the data created. Which user can perform this operation will depend purely on the application.

Delete – There should be a way to remove the data created. Again like Update, who can perform delete operation will vary depending on the application.

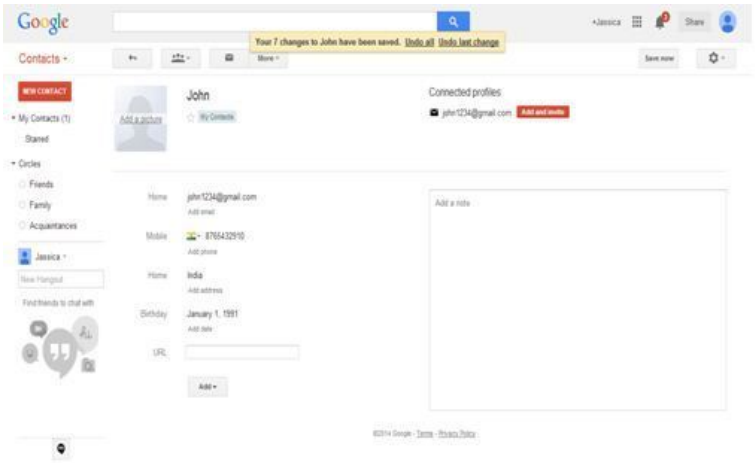
Remember that Application should follow CRUD, but not CURD

ie, Create will happen first, Update and Delete should happen only after the data is Read.

Create

Let's take example of well-known website Google. When you first time register or create an account in this website, you have to fill certain details and click on register or sign up button. On clicking on the button, information entered by you will be verified and added into the database and your account will be created. This is one of the example for CREATE operation.

Similarly, adding a contact in gmail application of your Google account is also an example of create operation.



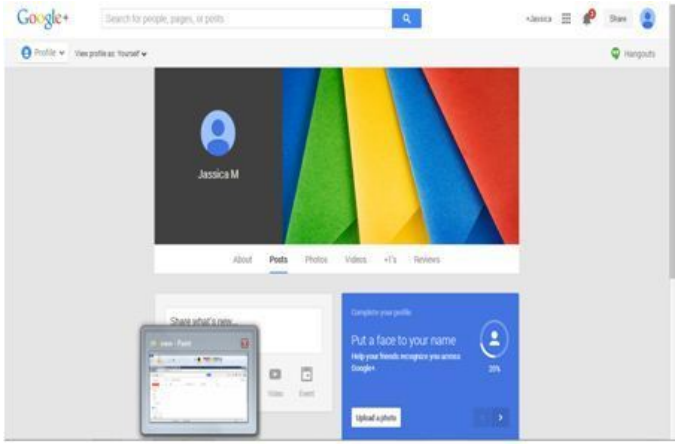
You can find these kind of options in many other websites. Creating your Facebook, twitter, IRCTC account, registering in naukary.com, even creating an event in your Facebook account are all examples of create operation.

Whenever a new set of information is added into the database of the system, then it is a create operation.
It results in creation of new entry into the database of that web application.

Read

Let's take the example of Google again. When you log in to your Google account, a home page appears. It is a read operation. In this, the web application retrieves the information from the application's database, corresponding to the log in credentials which you have provided and displays it.

Viewing your profile by clicking on view profile is also an example of read operation.



When you are searching a person by their name, it is also a read operation. In this the web application will search for various user profiles with that particular name and displays the list.

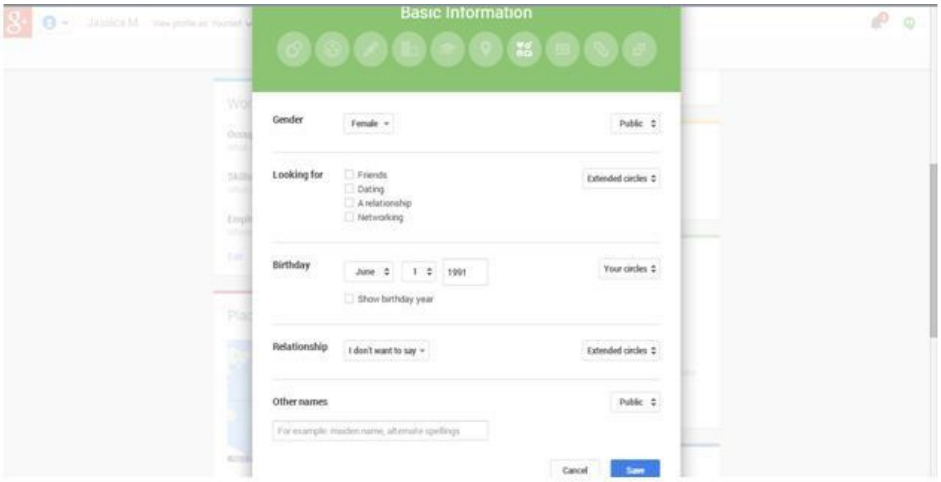
Even viewing your contact list of gmail is an example of read.

You can relate these examples with options being provided by many other social networks and web applications.

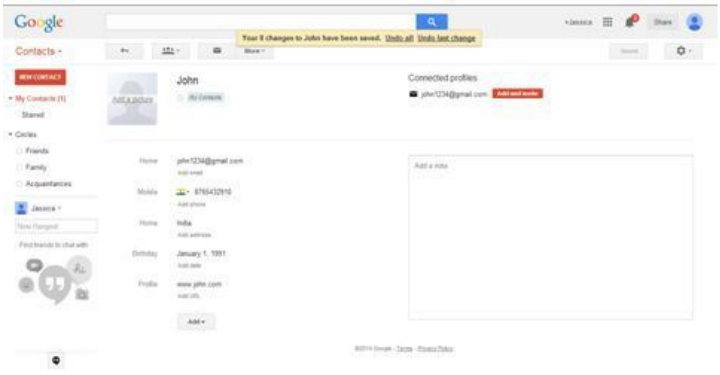
It shows that, read is basically, retrieving data from the database and displaying it to the user.

Update

Think, you want to change some of your basic information in your Google account, then, you will click on the edit button, which will take you to the edit profile page. It will display all the information which you have already added. You can enter the new information and by clicking on save, the information will get updated. Old information gets overwritten by the new information in the database.



Editing the contact details in gmail account is another example of update operation.

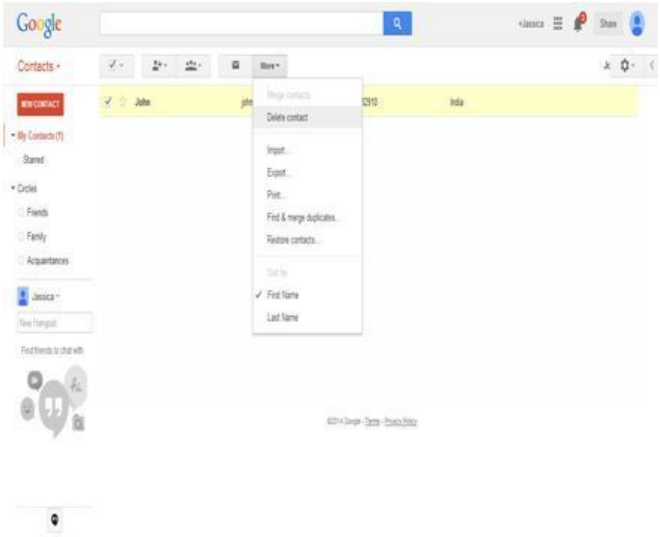


First time while filling the information, you might have left few non-mandatory fields blank. Filling those information later is also an example of update operation. Even removing few information (not all) like, phone number from your profile is also an example of update. Here the particular information will be set to blank.

Basically, update is, first a read operation, which displays the information which is already present in the database of the web application and then overwrites it with a new information.

Delete

Let us take the example of contacts in gmail. If you want to delete a contact from the contacts list, you can select the particular contact and click on the delete button. It will delete the contact from the contacts list.



If in future, you do not wish to use the account which you have created, then you can go for deactivation of the account. This is also an example of the delete operation.

Now a day most of the websites do not provide direct deletion, they allow you to just deactivate the account. Few websites even have facility to delete the account.

Delete is basically, deleting the information from the database of the website or making the information inaccessible.

Ask a doubt *(Misuse of 'Ask a Doubt' Section will be dealt as per the Terms & Conditions of Campus Commune)*

Open Doubts | **Closed Doubts**

GANGISETTY VS CHANDANA

Ithink its update oeration performed Whenever a new set of information is added to the database of the sysytem ,as we are updating the data base with new features but its given as create what is the right answer?

about 26 days ago

YUKTI **Best Answer**

Whenever we make any changes to already existing information that is referred to as an UPDATE,while adding some new information is CREATE.

SUMANTA SAHA

UPDATE is used to modify a single record in a database not the whole database. UPDATE is not applicable for the whole database. In order to insert a new record(which you meant update) CREATE operation must be performed.

SUMAN SINGH

When you are creating an entry, you are simply adding a new information or a new entry to the database. Where as updating refers to changing the values of existing entries.

RIYANKA DIWAKAR

How many courses are there in java lounge?

about 1 month ago

KARTHIK ND

Why most of the websites are not providing direct deletion?

about 2 months ago

ALAMELU N

Without reading(READ) the information that you want to delete,you cant delete(DELETE) the information.So delete cannot be performed directly !

PAVAN K

If you perform the direct deletion that content in your account for suppose is a redundant one then it will cause a user dilemma in selection of content so a read and the next step of deletion makes the best output and its a reliable method of deletion

UDAYADITYA BORUAH

The point of not providing direct deletion makes the account created by the user to be again accessible in the nearby future if he/she wishes to. This prevents repeated space allocation to a single user in the database, thereby preventing complications, if any.

SUMANTA SAHA

PAVAN K, could you please explain about the redundat data you mentioned?

Java

☐ Course Completion Quiz

2.Need of Programming and Introduction to OOP approach

Objective

- Introduction to problem solving approach
- Why problem solving
- Step by step approach to problem solving
- Key elements in problem solving
- To introduce programming as a means to automate a solution to any problem
- To understand the term 'Object-Oriented Programming'
- To know about Classes and Objects in the context of Object-oriented programming
- To introduce programming as a means to automate a solution to any problem
- To understand the term 'Object-Oriented Programming'
- To know about Classes and Objects in the context of Object-oriented programming

2.1 Introduction to problem solving approach

2.1. Introduction to problem solving approach

Most people spend most of their time, at work or at home, solving problems. Most problems we face are small, some are large and complex, but they all need to be solved in a satisfactory way.

let's take a few moments to understand what we mean by a problem and problem solving.

A problem is a difficulty or challenge, or any situation that needs a solution

Driving a car to office, Making a cup of tea, Solving a crossword puzzle . all these are examples of problems that we face every day.



Why Problem Solving

Problem Solving is actually a skill,and there are no universal methods to be followed to solve a problem. Basically there could be multiple solutions to a single problem and one must try all the possible solutions one by one until we come across with the correct and feasible solution.

Ask a doubt *(Misuse of 'Ask a Doubt' Section will be dealt as per the Terms & Conditions of Campus Commune)*

2.2. Step by step approach to problem solving

Effective problem solving involves number of steps or stages. They are

i) Problem Identification

This stage involves detecting and recognizing that there is a problem; identifying the nature of the problem; defining the problem. This phase also involves , structuring the problem. Structuring the problem is all about gaining more information about the problem and increasing understanding. By spending some time defining the problem will helps to understand it more clearly and it will leads you to the next phase.

ii) Identifying possible solutions and Making Decisions

Once you complete the information gathering in the first phase, the next step is to identify all possible solutions. Once the requirements have been finalized, we design a solution(s) to the issue. We might note the steps on a piece of paper or we might take a print out of it. Referring to these artifacts makes our work simpler and gives us a clear idea of how to proceed with a task .Hence design becomes a very important task as far as any system development is concerned.

iii) Implementation

Implementation means acting on the chosen solution .This phase deals with the actual process of implementation of our design. In terms of programming, the code would follow the design and then write code to translate the human-understandable design to a machine understandable language. During implementation more problems may arise especially if identification of the original problem was not carried out fully.

iv) Test the outcome – Testing

The outcome of our implementation needs to be tested before it is actually deployed. Similarly, all code needs to be thoroughly tested to make sure that bugs are not present in it. If any bugs are identified, they need to be removed and the system needs to be checked again for consistency. Hence testing is a very important phase of problem solving.

v) Deployment and Maintenance

The developed and tested outcome is finally given to the customer. The deployment phase makes sure that the software that has been developed works fine in the customer's environment as per the customer's requirements. Maintenance phase deals with the periodic updates that might be required for the application.

Example

Let us take an example of making a cup of tea. This is a minor problem but a problem nevertheless. As a human being , we normally follow the established process of making tea. Let us assume that we need to automate the system of making tea and we are building a tea-vending machine. In the beginning, the tea-vending machine does not know what it needs to do and what will be the outcome of its actions. Hence we program the machine to understand our requirements and work accordingly. Assume that we are going to instruct the machine to make tea. We need to think of the following things

1. What should be provided as the ingredients
2. What should be done with the ingredients
3. What will be the outcome

Ingredients of tea will normally be milk,water, sugar and tea-leaves. Hence we instruct the machine to take milk, sugar and tea-leaves as the input.




Now once we have the inputs, we need to take the following steps

1. Boil water
2. Add tea-leaves
3. Add sugar
4. Add milk

We will instruct the machine to perform these steps sequentially. The machine understands this and using the inputs that were provided to it, will solve the problem.

Finally the outcome of these steps will be tea and in the machines terms, this is called as the output.

Ask a doubt *(Misuse of 'Ask a Doubt' Section will be dealt as per the Terms & Conditions of Campus Commune)*

Open Doubts | **Closed Doubts**

ANKITA SHEKHAR

What is design or a plan? Code, document, blueprint or a pictorial outline?

about 23 days ago

MOHIT SRIVASTAVA

Design is basically the outlay that how we are going to tackle the problem, let me quote an example if a bowler is bowling a bouncer then the batsman have some shots to tackle this kind of delivery, these so called "some shots" here is a plan of facing a bouncer ball.

now coming to your question it can be document, pictorial outline(flow chart) or a code(module functionality will be defined)

VELANTISH V. **Moderator**

Design is *a roadmap or a strategic approach for someone to achieve a unique expectation. It defines the specifications, plans, parameters, activities and processes.*

Let us consider an anology of constructing a building, before even starting with the actual construction we have a blue print (called the plan i.e nothing but the design) which would state whether the contructed house is 2BHK or 3BHK. Where is the entrance of the room and passage. Because if there are any modifications to be done at that point in time it can be done easily with less cost and time. Where as just imagine the to be done after/during construction there there would be lot of cost, time and effort incurred. Similar is the case with software development and the importance of design.

Design consists of components, classes and sequence in which it has to be implemented to arrive at the results. This is represent by UML (Unified Modelling Language).

PRIYANKA SAWANT

What is the right ans???

about 25 days ago

MOHAMMED RAHIMAN

question:: what is done in 4th step of problem solving??

I answered TESTING.....is it wrong answer????

similarly ,what is done in 3rd step of problem solving???

I answerd CODING...is it really wrng answer?????

about 2 months ago

BHAVYA PULIPATI

I also faced the same problem

PRABHAKARAN

It seems that in the question answers, they have included a step in between Identifying possible solutions and Making Decisions and implementation namely testing the design. One of the reference links also has it like that.

PRIYANKA SAWANT

faced the same problm ..fourth step is implementation..is also wrong

SUVIGYA MITTAL

so whats the correct ans???

SAI KANUPARTHI

Same problem for me too.. This question repeated for me twice and second time I have not attempted it so to escape from negative marking.

SOURAV CHAKRABORTY

in one of aspire modules, they listed seven steps of problem solving...its confusing

MADDI

Third step is **implementing the solution**.... not just coding.

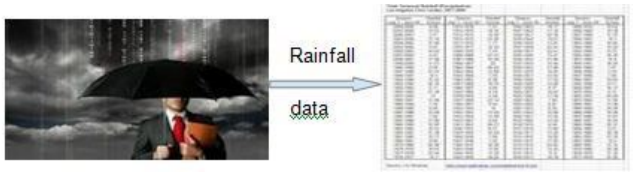
2.3. Key elements required for problem solving

Good Analytical Skills

Analytical skills are the ability to imagine, collect information, arrange it, analyse, solve complex problems, and make decisions. The ability to organise data and draw proper correlations and then interpreting these trends in terms that are meaningful to others is called Analytical skill.

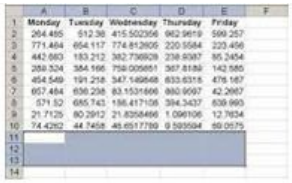
Let us look a example to understand the good analytical skills

i) **Rainfall Data for past thirty years at random are supplied**

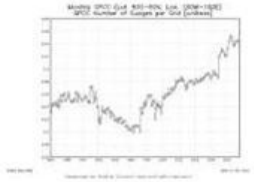


ii) **Organize and arrange the data**

By feeding it to excel sheet or any logical approach



Draw graphs to demonstrate the data



iii) **Output**

Make reasonable predictions about the extent of rain next year.

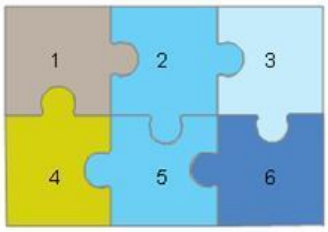
This is an example of using analytical skills.

Declarative and Imperative Knowledge

i) **Declarative knowledge** is factual knowledge or the knowledge of the solution. For example knowing that "A pen is used for writing" is a declarative knowledge. Declarative knowledge is knowledge or the possession of information that is either true or false. It describes objects and events by specifying the properties which characterize them; it does not pay attention to the actions needed to obtain a result

Take an example of a jigsaw puzzle

We know that the solution of this problem is



so this knowledge of the outcome is know as declarative knowledge.

ii) Imperative Knowledge

Continuing with the above example, although we know the result, we also need to know how to reach the result. This 'how-to' knowledge is called as imperative knowledge. Let us consider the steps

1.Arrange the pieces according to their number

2.Turn the pieces fitting to nearby pieces.

3.Join the pieces

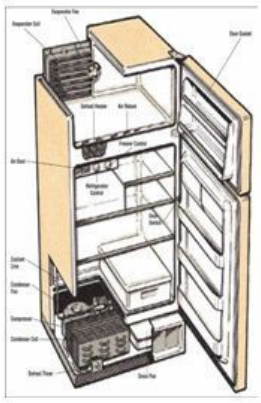
As we can see, if this order is followed, the puzzle would be solved. Hence the knowledge of the steps to reach the solution is called as Imperative knowledge.

Abstraction

A representation or model that includes the important, essential or distinguishing aspects of something while suppressing or ignoring less important, immaterial or diversionary details is called as Abstraction. Hence abstraction is a way of managing problem complexity by focusing on the essential details and ignoring lower level details.

An example can be taken of a refrigerator

As a user, we just need to know that a fridge can be used to preserve food materials by freezing them. Hence the user just needs to open the fridge and put the items inside.



User does not need to know the internal functioning of the fridge, hence the wiring,electronics and other details are hidden behind the body of the fridge. This process of hiding the irrelevant data is called as Abstraction.

Abstraction types

i) Functional Abstraction

Let see the above refrigerator example, all refrigerator contains a freezer box which can preserves the food material below zero degree Celsius temperature. As a user we need to get the food material to be preserved but we don't need how it will be get preserved , this way of hiding
irrelevant details or process and getting the functionality of preservation to be done is an example of functional abstraction.

ii) Data Abstraction

Let see the above refrigerator example,the inner temperature of the refrigerator is an essential data which is sensed by the sensor and refrigerator monitors and controls the inner temperature. Here inner temperature is the internal data which is used by the refrigerator to maintain the temperature of the fridge, which is not controlled or altered by external mechanism and is used by the functionalities of the refrigerator is an example of data abstraction.

Knowledge in programming language

```
def add5(x):
   return x+5

def dotwrite(ast):
   nodename = getNodename()
   label=symbol.sym_name.get(int(ast[0]),ast[0])
   print '    %s [label="%s" % (nodename, label),
   if isinstance(ast[1], str):
      if ast[1].strip():
         print '= %s"];' % ast[1]
      else:
         print '"]'
   else:
      print '"];'
      children = []
      for  in n, childenumerate(ast[1:]):
         children.append(dotwrite(child))
      print ,'    %s -> (' % nodename
      for  in :namechildren
         print '%s' % name,
```

A programming language is a computer language programmers use to develop applications, scripts, or other set of instructions for a computer to execute.

Programming languages can be used to create programs that control the behaviour of a machine and/or to express algorithms precisely.

Thousands of different programming languages have been created, mainly in the computer field, with many more being created every year.

Elements of a programming language are -

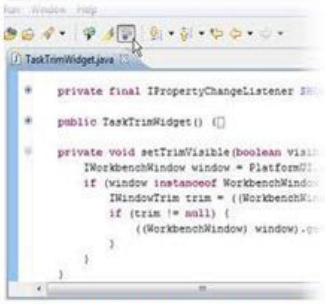
i) **Syntax:** structural elements of the language. Programs must be syntactically correct.
Eg: English as a language has got a certain syntax. We have predefined syllables and words like 'a' or 'cup' or 'welcome'. These words convey a meaning to the language.

ii) **Grammar:** defines how syntactical elements need to be combined to form programs
Eg: Continuing the same example, English language has got a certain grammar. The grammar has to be followed to construct a proper sentence. 'I is a boy' is grammatically wrong although syntactically correct.'I am a boy' is syntactically and grammatically correct.

iii) **Semantics:** defines meaning of the code
Eg: Continuing the same example, English language has got a certain semantics. 'An elephant is white and has got two legs' is syntactically , grammatically correct. But it doesn't make any sense. Hence it is semantically wrong.

Good programming practices

i) Good vs Bad programming



proper variables, aligned properly

unused variables, not aligned

Superior coding techniques and programming practices are hallmarks of a professional programmer. The coding techniques are primarily those that improve the readability and maintainability of code,whereas the programming practices are mostly performance enhancements.

ii)

The readability of source code has a direct impact on how well a developer comprehends a software system.

iii) Maintainability

Code maintainability refers to how easily that software system can be changed to add new features, modify existing features, fix bugs, or improve performance

Although readability and maintainability are the result of many factors, one particular facet of software development upon which all developers have an influence is coding technique. The easiest method to ensure that a team of developers will yield quality code is to establish a coding standard, which is then enforced at routine code reviews.

Ask a doubt *(Misuse of 'Ask a Doubt' Section will be dealt as per the Terms & Conditions of Campus Commune)*

2.4. Evolution of Programming Approaches

As we have seen in the previous course, for automating the solution of a problem, we need to do programming. Originally when the programming evolved, the adopted approach was small with complex syntax and machine code. With time, as requirements increased, the logic became much more complex with increase in the program size.

The following are the different programming approaches which evolved were:

- Unstructured Programming
- Structured Programming
- Procedural Programming
- Modular Programming
- Object Oriented Programming

The initial evolution methodologies had the following disadvantages

- Large volume of code and logic
- Lack of information hiding
- Poor readability

- Non-maintainable
- Reduced reusability

Ask a doubt *(Misuse of 'Ask a Doubt' Section will be dealt as per the Terms & Conditions of Campus Commune)*

2.5. Elements of Programming

For programming to be done, we need the following

- What is the information that is to be processed in the problem - Data
- What are the types or ways to process the information - Logic
- How will the processing happen using the logic – Methodology

Structured Programming Approach

Structured approach to programming, implements a system as a collection of functions that accepts data, processes it and returns an output. A function sometimes calls another function, when it wants to make use of the functionality defined in the other function. The flow of program execution is based on the structure of the program written. There is more dependency between the variables and the program like a chain.

Example: C language

The way of structured programming doesn't relate to real world problem scenarios. The current problem scenarios doesn't call for functions that operate on data, rather the scenarios have data and their associated behaviour. In such cases, the structured way of programming becomes very huge and cumbersome. Information has to be manipulated across all functions and an association of varied information in a problem scenario is not

possible.

Object Oriented Programming Approach

We write programs that make up software to solve a problem in real world. In real world problem domains we do not find any functions that work on data. We find objects that are described by some state (data) and having some behaviour (functions or routines).

Object Oriented (OO) Approach to programming is about implementing a system as collection of objects that have state and behaviours, interacting with each other to achieve expected functionalities.

Example: C++ & Java

In OO programming, the program basic entity is object. An object is any real world entity that has a well-defined structure and behaviour.

Forget programming for a while. Think about the Real World and the things that are in it. What things are objects? What things are not objects?

Look around you. You may try to list five objects and list five non-objects.

	Objects	Non-Objects
1		
2.		
3.		
4.		
5.		

	Objects	Non-Objects
1.	Pen	The upper 37% of pen
2.	Computer Keyboard	The air above the keyboard
3.	Shoe	Colour of the shoe
4.	Mouse	Sound of the mouse click
5.	Car	Speed of the car

It is easier to list things that are objects than to list things that are not objects. Just to talk about something seems to make it an object, somehow. When we observe, the humans view the world in object-oriented terms. The human brain wants to think about objects, and our thoughts and memories are organized into objects and their relationships. Perhaps non-human brains work differently.
One of the ideas of object-oriented software is to organize software in a way that matches the thinking style of our object-oriented brains.

Look at my list of objects and your own list and try to describe what all objects have in common. What makes an object?

- An object is made of tangible material (the pen is made of plastic, metal, ink).
- An object holds together as a single whole (the whole pen, not a fog).
- An object has properties (the color of the pen, where it is, how thick it writes.).
- An object can do things and can have things done to it.

The first item in this list is too restrictive. For example, you can think of your bank account as an object, but it is not made of material. (Although you and the bank may use paper and other material in keeping track of your account, your account exists independently of this material.) Although it is not material, your account has properties (a balance, an interest rate, an owner) and you can do things to it (deposit money, cancel it) and it can do things (charge for transactions, accumulate interest).
The last three items on the list seem clear enough. In fact, they have names:

i. An object has identity (each object is a distinct individual).
ii. An object has state (it has various properties, which might change).
iii. An object has behavior (it can do things and can have things done to it).
iv. This is a somewhat ordinary description of what an object is like. (This list comes from the book Object-oriented Analysis and Design, by Grady Booch, Addison-Wesley, 1994.) Do not be surprised if other notes and books have a different list. When you start writing object-oriented software you will find that this list will help you decide what your objects should be.

Every object has its own characteristics. Every object has 5 sets of characteristics Viz. State, Behaviour, Responsibility, Communication and Identity.

Ask a doubt *(Misuse of 'Ask a Doubt' Section will be dealt as per the Terms & Conditions of Campus Commune)*

2.6. Characteristics of an object

The five characteristics of an Object are explained below:

States of an Object: The states of an object are known by the various attributes that describe the Object. Example: If Human is an Object, colour of Hair, number of Teeth, skin tone, etc., describe the object. These form the states of the Object, Human

More Examples:

i. Make of Tyres, Colour, Engine Type, Mileage of a Car Object.
ii. Account Number, Account Holder Name, Deposit Amount, Transaction Details in a Bank Account Object.
iii. Employee ID, Employee Name, Date of Birth, Date of Joining, Number of Dependants in an Employee Object.

Behaviours of an Object: The behaviours of an Object is known by its outcome or reaction whenever there is a change in the states of it. The changes can also be infused by operations which are performed on the states of the object.

Examples:

- Add, Update, Remove dependant information of an Employee object

- Calculate performance of the Car based on engine type and mileage
- Print transaction details in a Bank Account

Responsibilities of an Object: The responsibilities of an Object are of two types namely Knowing Responsibility and Doing Responsibility

? Knowing Responsibility – Knowledge of the states of the Object

? Doing Responsibility – Operations performed on the states of the object

Examples:

The knowledge of the states like knowing the dependant details, retrieving date of birth of an employee are knowing responsibility.

All operational logic performed on the states of the object through its behaviours constitutes the Doing responsibility. Example: Calculate performance of the car

Communication between Objects: In a typical problem scenario, several objects communicate with each other to perform operations, send messages and return data.

Example: Employee object has a salary account in a bank. So both objects communicate with each other to perform all bank related transactions of the employee.

Identity of Objects: Identity of objects as explained earlier relates to the distinctness of the object. We identify the objects in a problem scenario based on all nouns of the project. The behaviours are identified using the verbs operated upon by the nouns

Example: Employee updating Bank Account information to Corporate Portal.

Distinct Objects: Employee, Bank Account, Portal

Behaviours: Updating the related information among the objects.

To enable understand the characteristics of an object, let us consider the following example

Consider a tube of four yellow tennis balls.

- Is the tube of tennis balls an object?
- Is each tennis ball an object?
- Could the top two balls be considered a single object?
- Is the color of the balls an object?
- Is your understanding of tennis balls an object?

Answers:

Is the tube of tennis balls an object?

Yes. It has identity (my tube of balls is different than yours), it has state (opened, unopened, brand name, location), and behavior (although not much).

Is each tennis ball an object?

Yes. It is OK for objects to be part of other objects. Although each ball has nearly the same state and behavior as the others, each has its own identity.

Could the top two balls be considered a single object?

Not ordinarily. Each has its own identity independent of the other. If they were joined together with a stick you might consider them as one object.

Is the color of the balls an object?

No. It is a property of each ball.

Is your understanding of tennis balls an object?

Probably not, although it is unclear what it is. Perhaps it is a property of the object called "your brain."

Software Objects:

Many programs are written to do things that are concerned with the real world. It is convenient to have "software objects" that are similar to "real world objects". This makes the program and what it does easier to think about. Software objects have identity, state, and behavior just as do real world objects. Of course, software objects exist entirely within a computer system and don't directly interact with real world objects.

An object stores its state in fields (variables in some programming languages) and exposes its behavior through methods (functions in some programming languages).

What are software objects made out of?

Computer Memory.

Its important to ask yourself :

- **What possible states can this object be in?**

- **What possible behavior can this object perform?**

Think your table lamp may have only two possible states (on and off) and two possible behaviors (turn on, turn off), but your table radio might have additional states (on, off, current volume, current station) and behavior (turn on, turn off, increase volume, decrease volume, seek, scan, and tune).These real-world observations all translate into the world of object-oriented programming.

Ask a doubt *(Misuse of 'Ask a Doubt' Section will be dealt as per the Terms & Conditions of Campus Commune)*

Open Doubts | **Closed Doubts**

MOUNIKA

In to how many phases can a checked algorithm can be programmed?

about 28 days ago

SAI KANUPARTHI

I think it depends on the number of steps in the algorithm.

PARIMALA PITCHIKA

Can I have examples of software objects?? Is the entire purpose of a program revolve around object?

about 2 months ago

HEMANKITA

For suppose if you have a class employee and you access it using an object e1 then e1 is a software object here.

HEMANKITA

For suppose if you have a class employee and you access it using an object e1 then e1 is a software object here.

LAKSHMINADHA PRASAD THOTA

I will Explain this with Example, Cat is the animals we can see in the real world.

Cat c=new Cat(); here c refers the actual object of Cat i.e **new Cat().**

here new is a key word which creates new object every time.

The reference **c** is stored in Stack memory and the actual object **new Cat()** stored in Heap memory.by using this object reference we can access all attributes and methods of the class Cat. here class is a template and it acts as a factory of objects i.e we can create any no of objects from the class.

PARIMALA PITCHIKA

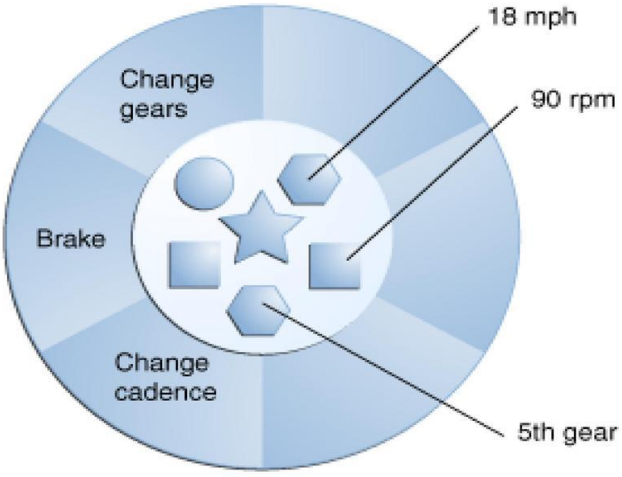
Thank u all of u

2.7. Classes and Objects

Consider this topic as introduction only. Detailed concept of class will be covered in following chapters. Hence consider code sample shared here for reference and not implementation.

Classes are generic representations of Objects in problem scenarios. They help in generalization of the object as a template. Each object becomes an instance or photocopy of this template.

Consider a car, for example:



Here Car is the template having the generic states and behaviours. The states and behaviour of the template are grouped in the following manner.

```
class Car {
  int speed = 0; //state
  int gear = 1; //state
 // Behaviour
  void changeGear(int newValue) {
       gear = newValue;
  }
 // Behaviour
 void speedUp(int increment) {
       speed = speed + increment;
  }
  void applyBrakes(int decrement) {
       speed = speed - decrement;
  }
void printStates() {
System.out.println("cadence:" +  cadence + " speed:" + speed + " gear:" + gear);
}
}
```

All various types of Cars which follow the generalization become Objects. Example: Maruti, Volkswagen, Audi, etc.,.

Basic difference between Programming Paradigms

Functional/procedural programming:

Program is a list of instructions to the computer

Object-oriented programming:

Program is composed of a collection objects that communicate with each other

Ask a doubt *(Misuse of 'Ask a Doubt' Section will be dealt as per the Terms & Conditions of Campus Commune)*

Open Doubts | **Closed Doubts**

RAJARAJESHWARI S

what is cadence in this program ? It is undefined , right??

about 19 days ago

ABHISHEK SRIVASTAVA

System.out.print() is defined in java.io package but why we can use System.out.print() without importing java.io package

about 28 days ago

PREETHI PATTABIRAMAN

The System object has references to `java.io.PrintStream` objects embedded in it. So you don't need to *explicitly* import these - the runtime can derive this information unambiguously since it was embedded at compile-time. Note also (in case there's any confusion), `java.lang` is implicitly imported, hence you don't require an import statement for `System` .

PALAK PARHI

what does the last function do?

about 2 months ago

MAHADEEP RAY

printStates function displays the values.

LAKSHMINADHA PRASAD THOTA

In last method **printStates()** having **System,out.println()** statement in this **System** is a predefined class available in **java.lang** package, **out** is a static variable available in System class of type **PrintStream** class, this is available in **Java.io** package and **println()** is the method available in **PrintStream** class, It is used to print/display the messages on the console.

VELANTISH V. **Moderator**

It displays in the console the current state of the car such as its speed and the gear(1st, 2nd or 3rd etc.) its running with.

2.8. Reference Links & videos

Reference links

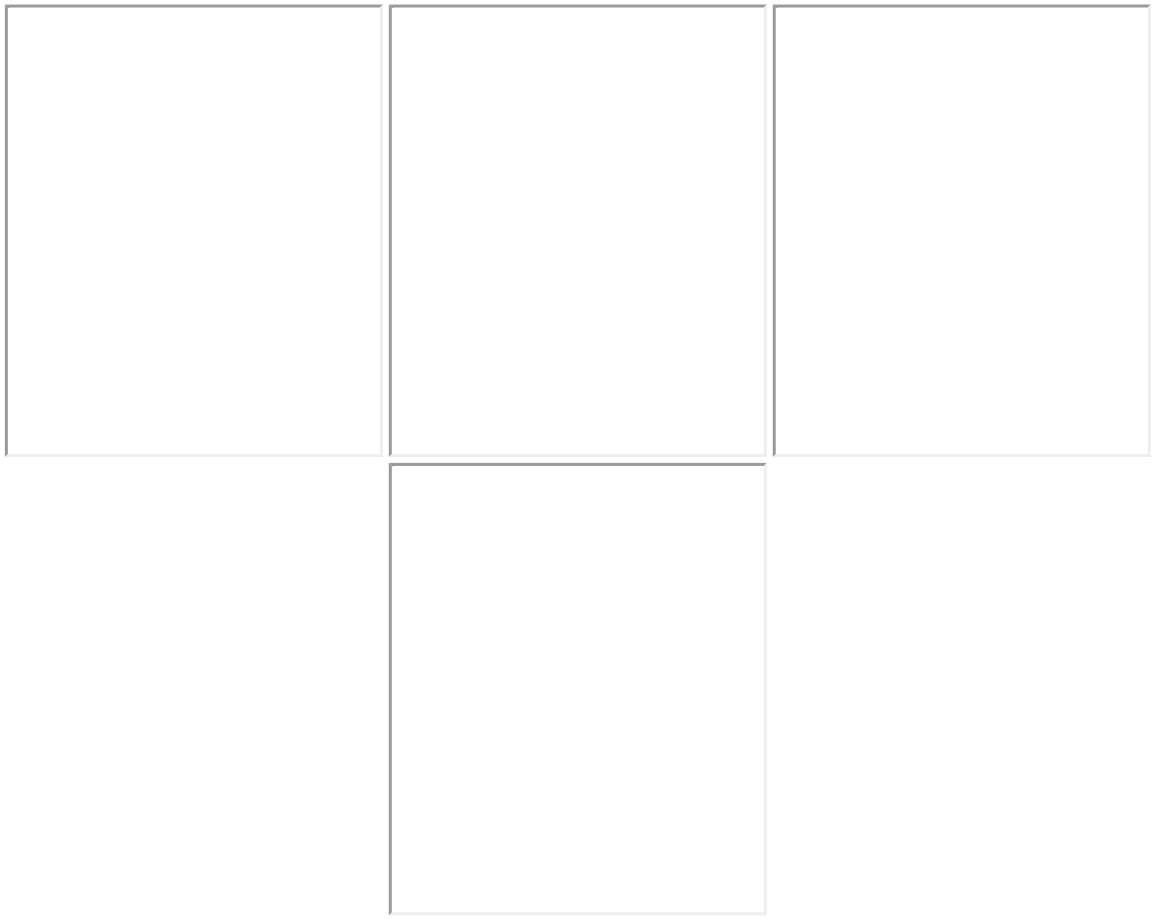
http://en.wikipedia.org/wiki/Problem_solving

http://www.mindtools.com/pages/main/newMN_TMC.ht

http://www.wikihow.com/Improve-Analytical-Skills

To know more on programming approach, visit http://www.maths.udsm.ac.tz/mujuni/mt512/Programming%20Approaches.pdf

Related Videos



Ask a doubt *(Misuse of 'Ask a Doubt' Section will be dealt as per the Terms & Conditions of Campus Commune)*

Open Doubts | **Closed Doubts**

SRIKAVYA PINGALI

Please help me out.....Marks are getting deducted even if I mark the correct option....!!!

End user requirements are understood in requirements analysis stage....I marked it as true....Marks got deducted.

In designing phase involves preparing document blueprint and pictorial design but not coding as per this material....In this also I have my marks.....!!!

There are 2 other questions in the way where I have lost marks....!!!

Please suggest me where should I report about this as same happened with me in aspire also...!!!!

about 1 month ago

Java

Course Completion Quiz

3.Introduction to Java,JVM,JDK

Objective

- Introduction to Java
- History of Java
- Brief to Java Platform, JVM, JDK, JRE
- Installation of JDK and its configurations
- Run a sample program

3.1 What is java

3.2 History of Java

3.3 JVM- Java Virtual Machine

3.4 JRE- Java Runtime Environment

3.5 Know more on Command Promt

3.6 Different TOOLS TO CREATE JAVA PROGRAM

3.7 Eclipse

3.8 Compileonline.com

3.9 Reference Links & videos

3.1. What is java

Java is a programming language created by James Gosling from Sun Microsystems (Sun) in 1991. The first publicly available version of Java (Java 1.0) was released in 1995.

Sun Microsystems was acquired by the Oracle Corporation in 2010. Now oracle is owning Java.

Java is Simple:There are various features that makes the java as a simple language. because Java is easy to learn and developed by taking the best features from other languages mainly like C and C++.

Java is Platform Independent-Java provides the facility to "Write once -Run any where"(Known as platform independent).

Java is Robust-Java has the strong memory allocation and automatic garbage collection mechanism. It carries out type checking at both compile and runtime making sure that every data structure has been clearly defined and typed. compiler checks the program for any error and interpreter checks any run time error that every data structure is clearly defined and typed.

Java manages the memory automatically by using an automatic garbage collector.

All the above features makes Java language robust.

Why Java is important to internet

In a network, two very broad categories of objects are transmitted between the server and your personal computer: passive information and dynamic, active programs. For example, when you read your e-mail, you are viewing passive data. Even when you download a program, the program's code is still only passive data. Even when you download a second type of object can be transmitted to your computer: a dynamic, self-executing program. Such a program is and the server yet initiates active agent on the client computer. For example, the server to display properly the data that the server is sending might provide a program.

As desirable as dynamic, networked programs are, they also present serious problems in the areas of security and portability

Let us begin by writing our first Java program that prints a message "Hello, world!" to the display console, as follows:

? Hello World program in java

Step 1: Write the Source Code: Enter the following source codes using a programming text editor (such as TextPad or NotePad++ for Windows or gedit for UNIX/Linux/Mac) or an Interactive Development Environment (IDE) (such as Eclipse or NetBeans - Read the respective "How-To" article on how to install and get started with these IDEs – to see more about IDE refer the Glossary).

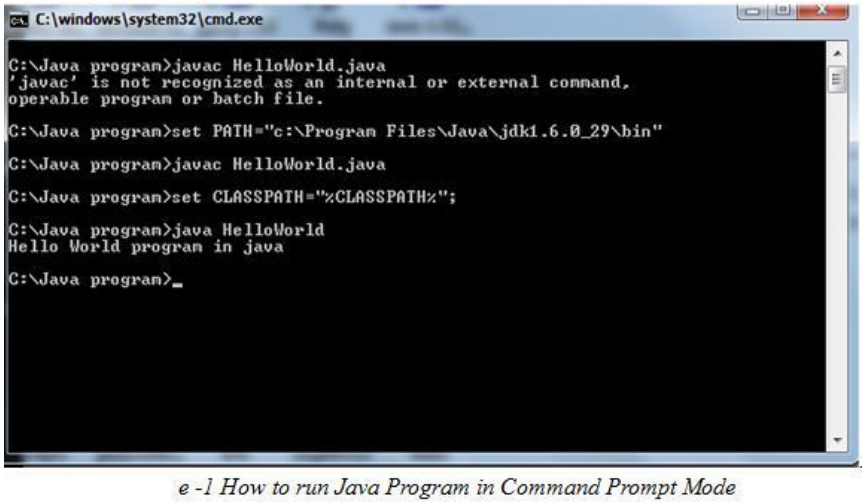
Save the source file as "Hello.java". A Java source file should be saved with a file extension of ".java". The filename shall be the same as the classname - in this case "Hello".

```
// Write simple Hello Program in Java

   class HelloWorld  //Start
  {
     public static void main(String args[])
        {

         System.out.println("Hello World program in java");
        }// Main methods ends
  }// Java class ends
```

Step 2: Compile the Source Code: Compile the source code "Hello.java" into portable bytecode "HelloWorld.class" using JDK compiler "javac". Start a CMD Shell (Windows) or Terminal (UNIX/Linux/Mac) and issue this command:

before run java file you should set the path and classpath
path: to instruct compiler that where your JDK is installed
How to set the path:
set PATH=.;C:\Program Files\Java\jdk1.5.0\bin(for example:
C:\Program Files\Java\jdk1.5.0\bin (Provided your jdk installed in the given path)

See the below screen shots in order to run any java program in command prompt mode



e -1 How to run Java Program in Command Prompt Mode

Summary:

Java is platform Independent and Portable.
Java is Robust and secured.
Java is pure object oriented language.

Java does compilation as well Interpretation.

Java latest version of JDK 7(Current version is being used in Industry)

JDK and JRK has to be installed in order to Java program.

Ask a doubt *(Misuse of 'Ask a Doubt' Section will be dealt as per the Terms & Conditions of Campus Commune)*

Open Doubts | **Closed Doubts**

SIVAKUMAR R

Is java really a pure object oriented language? If it is pure object oriented then its primitive data type should be objects, which is not. Am i right? Correct me if i am wrong.

about 9 days ago

HARIPRASAD R

I think Java is not pure object oriented language.

about 11 days ago

ABHISHEK SAH

I think there is a typo in **Step 2: Compile the Source Code:** Compile the source code "Hello.java" into portable bytecode "HelloWorld.class" .

It should be **Step 2: Compile the Source Code:** Compile the source code "HelloWorld.java" into portable bytecode "HelloWorld.class"

about 23 days ago

SAIF PATEL

Its correct,

because the mane of the program is "hello". So it will be compile by the name of the class..

on compiling it will generate the class file with the name of the class specified in the program.

MAYANK

it is incorrect because the main method is in HelloWorld class. so the name of file should be HelloWorld.java

SUBRATA MAITI

It is write because the class is not public, so you can name the file anything. If the class is public class then name should be as the class name. And always after compiling the .class should be same name as the classes present in the .java file.

SATYANARAYAN K

I tried to execute the given code in my PC.I have java (**jre7**) installed. I am unable to execute as shown in the above picture.Please help me out.Or do i need to install some other version of java and try it out?

about 29 days ago

THANGARAJ

install JDK7 and try it

KALYAN PILLA

Java Runtime Environment (JRE) is required to execute a java object file. In order to compile a java Source code you will need Java Development Kit (JDK).

KARMANJOT

name of the file "hello.java" is given wrong. try saving the file as "HelloWorld.java".

SATYANARAYANA MURTHY

can a compiler checks or detect run time errors?

about 1 month ago

LAKSHMINADHA PRASAD THOTA

Java compiler doesn't know/check the run-time errors or Exceptions..

SWATI SINGH

what is JRK? it also needs to be installed ?? plz reply

about 2 months ago

VEERABHADRASWAMY BODA

I think it's JRE not JRK . jre means java runtime enviroment which contains jvm,class libraries and other supporting files.It doesn't contain any development tools like compiler etc. so inorder to run your program which requires jvm and other class libraries jre must be installed in your computer

AMEYA TEMBHEKAR

What is a classpath and can you elaborate the declaration in the above program?

about 2 months ago

LAKSHMINADHA PRASAD THOTA

The CLASSPATH environment variable tells the Java Virtual Machine and other Java applications where to find the class libraries, including user-defined class libraries.

ABUL M

I cant understand about passive data, dynamic, active data clearly. Can anyone explain it with some other example

about 2 months ago

PARIMALA PITCHIKA

Passive data is you can only read pages .i.e.,static.

Active/dynamic having dynamic data ,filling forms and having hyperlinks which makes you turn to another pages.

3.2. History of Java

Java is released in 1995 as a core component of Sun Micro system's Java platform. In year 1991 they make a platform independent software called it Oak. and in early 1995 Java 1.0 is officially released. This is the first version of Java as JDK1.0. After the first version, many classes, packages are added in Java in future and new version's are released .

Java version releases

Current version of Java is Java SE7. The Key features in Java SE 7:

- Strings in switch Statement
- Type Inference for Generic Instance Creation
- Multiple Exception Handling
- Support for Dynamic Languages
- Try with Resources
- Java nio Package
- Binary Literals, underscore in literals
- Diamond Syntax
- Automatic Null Handling

Differ Version Available prior to Current Version (Java SE7)

Version	Year of Release	Description
J2SE 1.4	6-Feb-2002	This was the first release of the java platform developed under java community process as JSR 59. The Major Changes are- assert Keyword, Regular Expression, Exception chaining, Image I/O API for reading and writing images, Java Web start included.
J2SE 5.0	30-Sept-2004	New features added are- Generics, Metadata, Autoboxing, Enumeratations, Swing, Var args, collections static imports etc.
Java SE 6	11-Dec-2006	New changes are- Scripting Language support, Improved web service support trough JAX-WS , JDBC 4.0 support, support for pluggable annotations, JVM improvements including Synchronization and compiler performance optimizations, Garbaze collection algorithms

Java Platform

- JRE and JDK : See the JDK Section in detail
- Java Programming Language:
- Java Virtual Machines – See the JVM Section in detail
- Base Libraries
- Integration Libraries
- User Interface Libraries
- Deployment
- Java Deployment
- Tool Specifications
- JDK Tools & Utilities
- Platforms

Ask a doubt *(Misuse of 'Ask a Doubt' Section will be dealt as per the Terms & Conditions of Campus Commune)*

Open Doubts | **Closed Doubts**

ALURI REDDY

What is the purpose of diamond box in java?

about 18 days ago

VELANTISH V. **Moderator**

Diamond symbol in java is used for generics (<>). Generally collections in java can hold any type of object such as employee, department, suppliers. But practically we would need to narrow down collection to hold strictly objects of just a class. That is where generics comes into picture.

You can read more details while learning collections in java.

PILLI

what is meant by type inference for generic instance creation?????

and also about java nio package??

about 1 month ago

LAKSHMINADHA PRASAD THOTA

I answered these question in the below post

M REDDY

what is JDBC 4.0 ??

about 2 months ago

LAKSHMINADHA PRASAD THOTA

JDBC stands for **J**ava **D**ata**b**ase **C**onnectivity, which is a standard Java API for database-independent connectivity between the Java programming language and a wide range of databases(DBs).

Features of JDBC 4.0
==========

1.Auto- loading of JDBC driver class
2.Connection management enhancements
3.Support for RowId data type
4.SQL exception handling enhancements
5.SQL XML support
6.DataSet implementation of SQL using Annotations

PARIMALA PITCHIKA

What is Type inference for generic instance creation?

about 2 months ago

LAKSHMINADHA PRASAD THOTA

In general if we want to create generic type collection we can use below syntax

ArrayList<String> list = new ArrayList<String>();

but in java7 there is a future called Type inference in this we no need to specify the generic type String inside the dymond brakets in the right side of ArryList object creation, we can just specify in the left portion,This is also known as the diamond operator i.e

ArrayList<String> list = new ArrayList<>();

Exactly only the left side of the statement must specify the type arguments <type>, while the right side will be inferred from the left side.

S KIRTHIKA

what is nio packagev means????

about 2 months ago

LAKSHMINADHA PRASAD THOTA

Java NIO (New Input Output) is an alternative IO (Input Output)API for Java.The main difference between Java NIO and IO is that IO is stream oriented, where NIO is buffer oriented, IO is blocking, while NIO is non-blocking.

stream oriented means that you read one or more bytes at a time from a stream,They are not cached anywhere, we cannot move forth and back in the data in a stream.
If we need to move forth and back in the data read from a stream, we will need to cache it in a buffer.

Java NIO is buffer oriented approach.Data is read into a buffer,we can move forth and back in the buffer.

Blocking means, that when a thread invokes a read() or write(), that thread is blocked until there is some data to

read, or the data is fully written. The thread can do nothing else in the meantime.

Non-blocking mode enables a thread to request reading data from a channel, and only get what is currently available,if no data is currently available. Rather than remain blocked until data becomes available for reading,the thread can then go on and do something else in the mean time..

ALURI REDDY

Java NIO (New IO) is an alternative IO API for Java.
The differences between Java NIO and IO

1) IO is stream oriented, where NIO is buffer oriented.
2) NIO allows you to manage multiple channels using only a single (or fewer) threads, but the cost is that parsing the data might be somewhat more complicated than when reading data from a blocking stream using standard IO.

3.3. JVM- Java Virtual Machine

A Java virtual machine (JVM), an implementation of the Java Virtual Machine Specification, interprets compiled Java binary code (called bytecode) for a computer's processor(or "hardware platform") so that it can perform a Java program's instructions. Java was designed to allow application programs to be built that could be run on any platform without having to be rewritten or recompiled by the programmer for each separate platform. A Java virtual machine makes this possible because it is aware of the specific instruction lengths and other particularities of the platform.

JVM is:

A specification where working of Java Virtual Machine is specified. But implementation provider is independent to choose the algorithm. Its implementation has been provided by Sun and other companies.

An implementation Its implementation is known as JRE (Java Runtime Environment).

Runtime Instance Whenever you write java command on the command prompt to run the java class, and instance of JVM is created.

The JVM performs following operation:

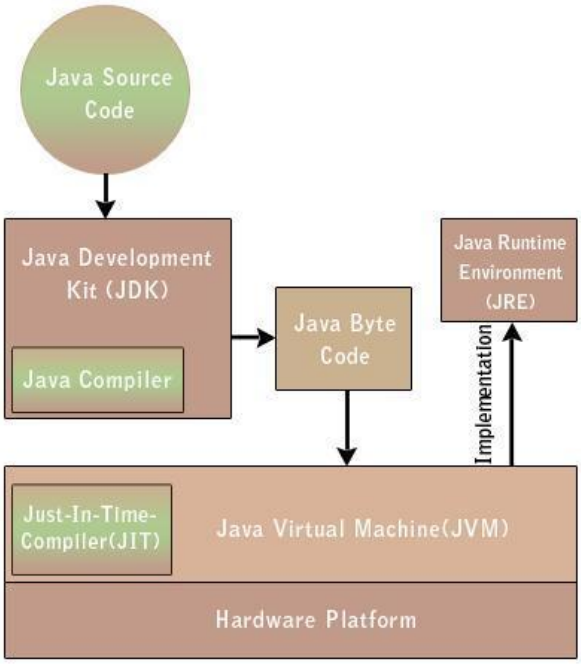
- Loads code
- Verifies code
- Executes code

- Provides runtime environment.

JDK - Java Development Kit

A Java Development Kit (JDK) is a program development environment for writing Java applications. It consists of a run time environment that sits on top of the operating system layer and also the tools and programming that developers need to compile, debug, and run applications written in the Java language.

The JDK (Java Development Kit) is used for developing java applications. The JDK includes JRE, set of API classes, Java compiler, Webstart and additional files needed to write Java applications. The JDK (Java Development Kit) contains software development tools which are used to compile and run the Java program. Both JDK and JRE contains the JVM



Ask a doubt *(Misuse of 'Ask a Doubt' Section will be dealt as per the Terms & Conditions of Campus Commune)*

Open Doubts | **Closed Doubts**

PERABATHINI KIRTI

Can anyone explain what "Both JDK and JRE contains the JVM" means?

about 19 days ago

SUMANTA SAHA

JRE contains JVM & JDK contains JRE. Hence both JRE JDK contains JVM.

ANURAG MODI

Is JVM a sort of document where only specification is given?

about 1 month ago

ABHISHEK SAH

No, JVM is the Java Virtual Machine where the compiled bytecode is run/interpreted. JVM makes the java programs platform independent. Sun has 5.5billion JVM enabled device. So a java program can be run in 5.5billion devices. JVMs are platform specific, and they provide same environment to the bytecode for all the devices.

SREE SETTURU

what is API?

about 2 months ago

MUTHU S

API is Application Programming Interfaces

LAKSHMINADHA PRASAD THOTA

Application Programming Interface (API), in the context of Java, is a collection of predefined packages, classes, and interfaces with their respective methods, fields and constructors. In Java, most basic programming tasks are performed by the API's classes and packages, which are helpful in minimizing the number of lines written within pieces of code. and it makes programmer's life easy to write code and it is reference for the programmer.

ALURI REDDY

An application programming interface (API) is a particular set of rules ('code') and specifications that software programs can follow to communicate with each other. It serves as an interface between different software programs and facilitates their interaction, similar to the way the user interface facilitates interaction between humans and computers.

TATA CONSULTANCY SERVICES

Java Lounge

Logout

Java

1.IT Application Overview And Features	Q
2.Need of Programming and Introduction to OOP approach	Q
3.Introduction to Java,JVM,JDK	Q
4.Operators and Variables in Java	Q
5.Introduction to class in java,access specifiers, this and static in Java	Q
6.Java Library, Packages, Use of import	Q
7.Conditional operations	Q
8.Basic Iterations and Arrays	Q
9.Designing Web Pages Part 1(HTML and CSS)	Q
10.Designing Web Pages Part 2 (JavaScript and JQuery)	Q

Course Completion Quiz

3.4. JRE- Java Runtime Environment

Java Runtime Environment (JRE), also known as Java Runtime, is part of the JDK. JRE provides the minimum requirements for executing a Java application; it consists of the Java Virtual Machine (JVM), core classes, and supporting files. Thus it covers most end-users needs and contains everything required to run Java applications on our system.

Java Environment Setup

Java installation and environment variable setup steps are clearly described in Video 6. Follow the instructions given in the video and set up the Java environment.

Download JDK 5 or above setup for installation. Link to download latest version of java, jdk 7 is given below.

http://www.oracle.com/technetwork/java/javase/downloads/jdk7-downloads-1880260.html

Set the environment variables CLASSPATH and JAVA_HOME after JDK installation.

Ask a doubt *(Misuse of 'Ask a Doubt' Section will be dealt as per the Terms & Conditions of Campus Commune)*

Open Doubts | **Closed Doubts**

SASIVINEETHA GATTUPALLE

Where can i find vedio 6?? There are no vediods in this submodule.Someone do respond please. Thank you.

about 25 days ago

SWATI PARIDA

what are core classes?

about 2 months ago

LAKSHMINADHA PRASAD THOTA

The classes which are available inside **java.lang** package are called core classes. because we no need to **import** any package to use these classes in the java program.

by default those classes and interfaces are available because **java.lang** package is a default package which is known by JVM.

example:- **System** class, **Math**

3.5. Know more on Command Promt

1. What is Command Prompt?

Command Prompt is a feature of Windows that provides an entry point for typing MS-DOS (Microsoft Disk Operating System) commands and other computer commands. The most important thing to know is that by typing commands, you can perform tasks on your computer without using the Windows graphical interface.

When you're using Command Prompt, the term command prompt also refers to the right angle bracket (>, also known as the greater than character) that indicates the command line interface can accept commands. Other important information, such as the current working directory (or location) where the command will be run, can be included as part of the command prompt.

For example, if you open the Command Prompt window and see the C:\> command prompt with a blinking cursor to the right of the right angle bracket character (>), you know that the command you enter will be run on the entire C drive of your computer.

2. How do I get to a command prompt?

Open the Command Prompt window by clicking the Start button Picture of the Start button, clicking All Programs, clicking Accessories, and then

clicking Command Prompt.

Execute a Sample Java Program using Command Prompt

Steps to run a java program is described in Video 6. We will provide you a sample Java program for execution.

Create a java file "TestClass.java" and put the sample code given below in it. We will study more on how to write a Java class later in this course. This is just to understand how to compile a Java program file and how to execute it using command prompt.

```
class TestClass {
 public static void main(String args[])
 {
   System.out.println("TestClass is running fine.");
 }
}
```

After execution of TestClass, you should get the message on the command prompt as "TestClass is running fine". Have you got the output on command prompt / console.

Ask a doubt *(Misuse of 'Ask a Doubt' Section will be dealt as per the Terms & Conditions of Campus Commune)*

Open Doubts | **Closed Doubts**

GOKULKUMAR V

Where can i find the link for video 6 mentioned above?

Thanks in advance

about 1 month ago

MAHIMA

Someone please provide the link to video 6.

3.6. Different TOOLS TO CREATE JAVA PROGRAM

Download & Install JDK:

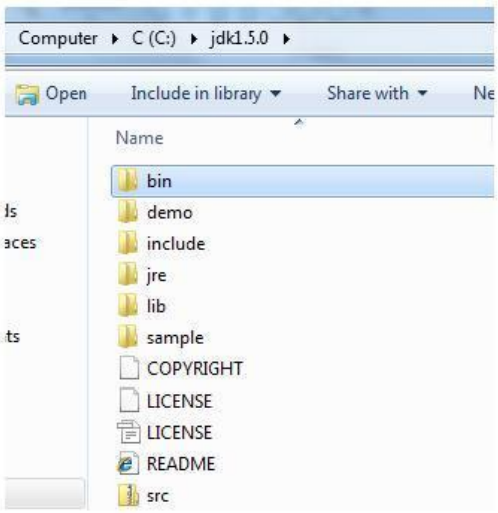
Oracle.com hosts the latest JDK installer and also the archives of older versions of JDK installer. The latest installer could be downloaded from the link:

http://www.oracle.com/technetwork/java/javase/downloads/index.html

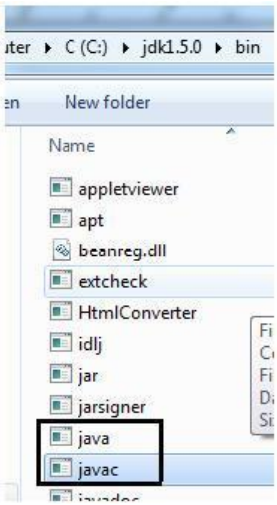
If you need a previous version of Java visit the following link:

http://www.oracle.com/technetwork/java/javase/archive-139210.html

After the download is complete, double click the installer and follow the installation process. Follow the recommended settings during installation.
Depending on the location that was chosen to install the JDK, you would see the following contents.
The outer most folder that has all the installed contents is generally referred to as Java Home.
So, as the per the below installation, the Java Home directory is C:\jdk1.5.0



The developer frequently needs an access to the java compiler (javac) and java interpreter (java). These tools come along with JDK and you will find them under the bin folder in Java installation folder.



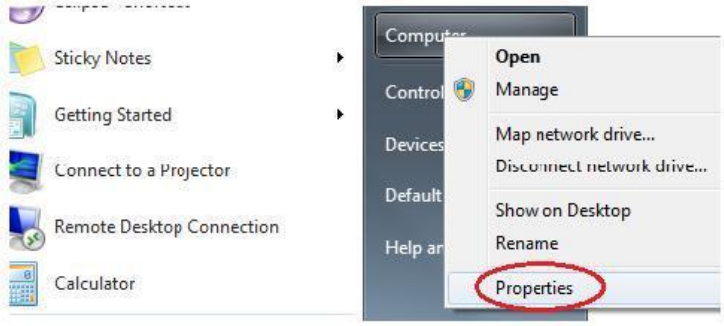
To enable to developer to use these tools from any directory in the command prompt, we set these values in the operating system environment variable.
Configure JDK:

Environment variables are key and value pair. PATH is one of the environment variable (key). It's value is a collection of more than 1 directory path. Operating system uses the PATH variable to scan all the directory when a user want to execute a tool without mentioning the installation path.

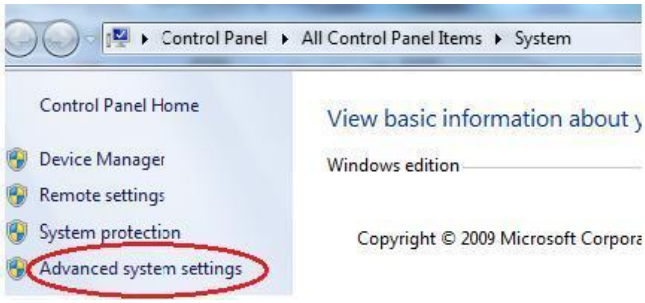
eg. PATH = C:\jdk1.5.0\bin; C:\Program Files\Intel\iCLS Client\; C:\Program Files\IBM\Trace Facility\

As you can see, the delimeter (;) is used to separate multiple directory mentioned as a value to the PATH environment variable.

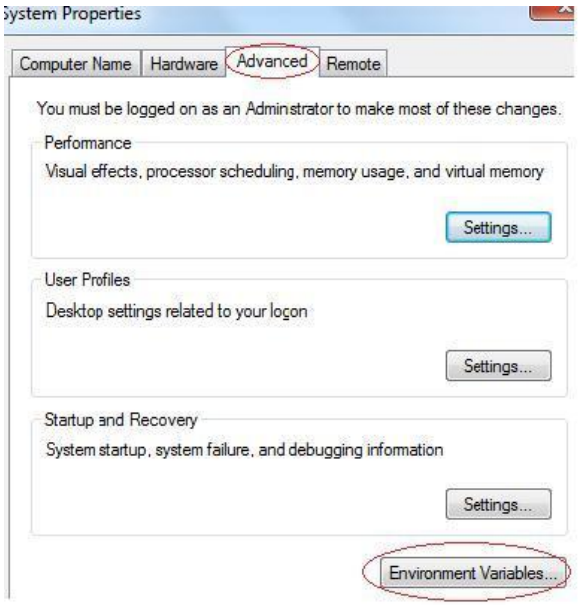
To set and environment variable, right click on Computer icon, and select Properties.



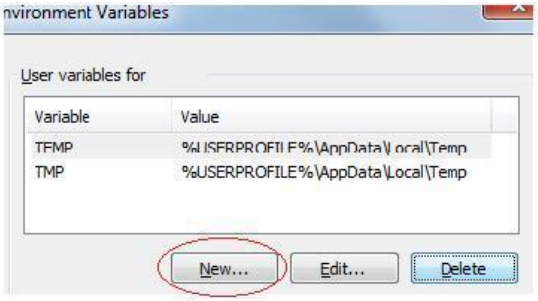
It will open Control Panel Home as shown below:



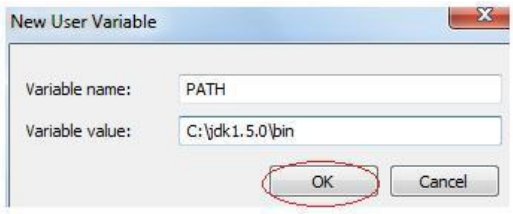
Right click on the Advanced system setting, it will open System Properties as shown below.



Go to the Advanced tab, and click on the Environment Variables button available at the bottom. This is would open a small window that will list all the existing environment variables. It has 3 button – New (to add a new environment variable), Edit (to edit the value of existing environment variable) and Delete (to delete and existing environment variable)

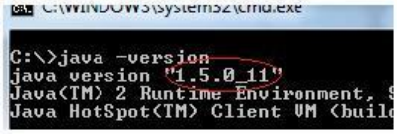


Click on the button – New. It will open a new window with 2 empty text boxes. Enter the variable name as PATH. In variable value, enter the path to the bin folder in java installation directory.



Click on OK to save changes.

You can verify the installation, by the command: java -version



Example:

Create a folder javaproject in your D: drive. Create a file MyFirstJavaProgram.java with following contents.

```
public class MyFirstJavaProgram
{
        public static void main(String a[])
        {
                System.out.println("Hello World");
        }
}
```

Open command prompt and browse to your folder D:\javaproject

Compile the file MyFirstJavaProgram.java with the help of the compiler – javac. The sytanx to compile a file with java extension is – javac FileName.java

If the compilation is successful, a file MyFirstJavaProgram.class is generated. The generated class file can be run with help of an interpreter – java. The syntax to execute a class file is – java FileName

Congratulations !! You have executed your first java program.

Ask a doubt *(Misuse of 'Ask a Doubt' Section will be dealt as per the Terms & Conditions of Campus Commune)*

Open Doubts | **Closed Doubts**

There are no doubts yet

3.7. Eclipse

Good as of now we have learnt how to install JDK on our systems. We also learnt how to write a java program, compile it and run the java program. Let us now learn another way of writing Java program. That is using an IDE. IDE means Integrated Development Environment. There are many IDE's available in the market but we will be using Eclipse. Eclipse is an Integrated Environment which provides programmers perform a lot of activities required during development at one place. Some of the activities that can be done using Eclipse are writing code, making code more readable, like indentation, providing spacing, compiling, running, debugging etc. There are many tools available in the market which helps in doing each of the activity mentioned above in separate windows, but IDE integrates all the above windows into one environment. Using Eclipse we can write programs in many languages like C, CPP and Java, html, Perl etc. But we will only be focused on learning how to use Eclipse to develop Java programs. First of all let us learn how to get eclipse installed in our system.

Install Eclipse:

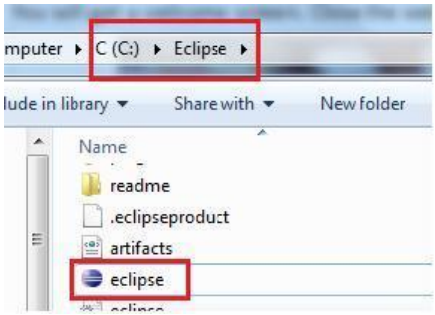
Eclipse has been evolving and there are many versions available and you may choose any of the existing versions for java development but we would recommend use of Galileo. You can download the same using the following url http://www.eclipse.org/downloads/packages/eclipse-ide-java-ee-developers/galileor . You may download the appropriate version depending upon your Operating System. Please refer the below diagram for the same.

Make sure you download the right installation file based on your OS (winodws, linux or Mac) and architecture (32 or 64 bit). You may download and place the archive (zip, tar.gz) at any location of the hard disk of your system. Then the archive needs to be extracted using any of the archiving tools provided by your OS. The installation of eclipse is done just by unzipping the downloaded zip file.

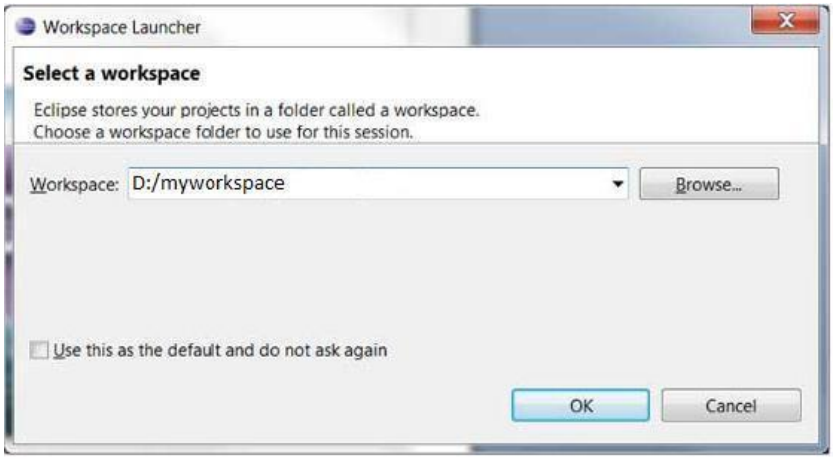
Configure Eclipse:

Before starting eclipse, make sure that you have downloaded Java Development Kit(JDK). We recommend you to download version 1.5. Refer to the download, install & configure JDK section present in this document for details.
Start eclipse from the eclipse installation directory. The eclipse installation directory will have an eclipse application as highlighted below.

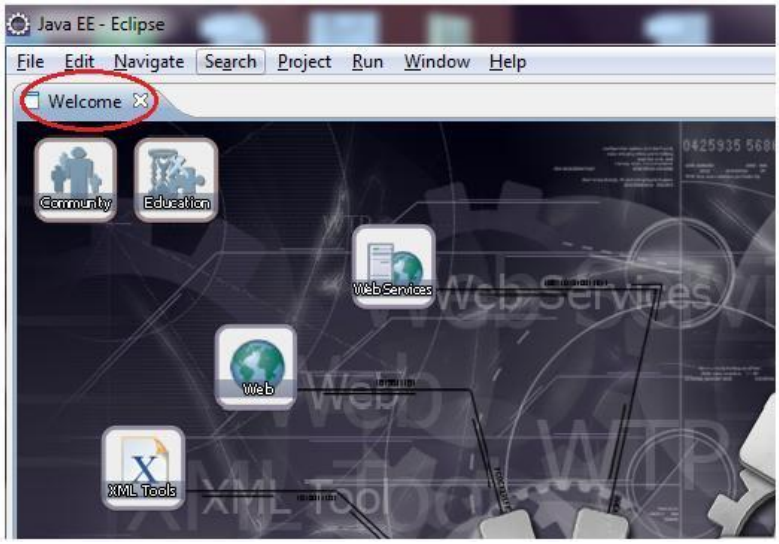
1. Start Eclipse



When eclipse starts up for the first time it prompts you for the location of the workspace folder. All your data will be stored in the workspace folder. You can accept the default or choose a new location.
If you check "Use this as the default and do not ask again", this dialog box will not come up again.



After selecting workspace, you will see a welcome screen. Close the welcome screen.

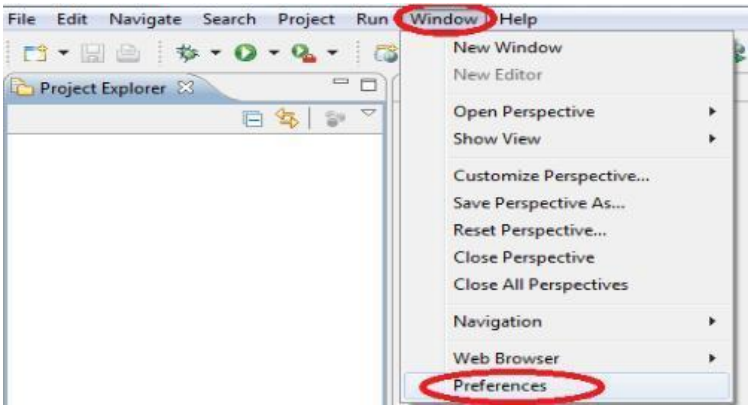


Setting up eclipse

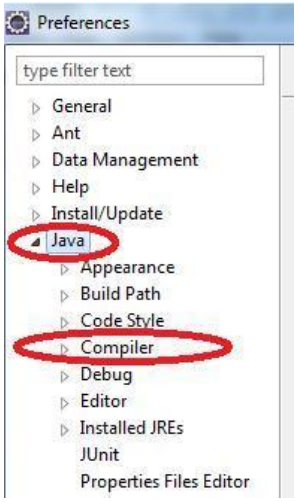
Now we need to make sure of certain things before proceeding with the creation of Java Project. We should set up environment for running programs in eclipse.

2.1 Compiler compliance level

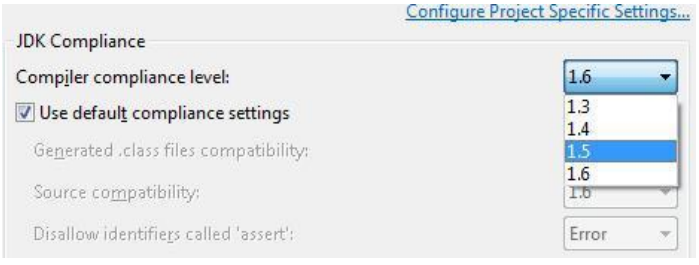
Right click Window and select Preferences.



Preferences window is opened. Now select Compiler from Java



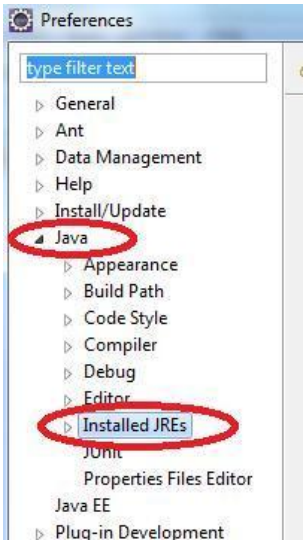
Compiler compliance level can be changed here. Select the appropriate version from drop down and click OK.



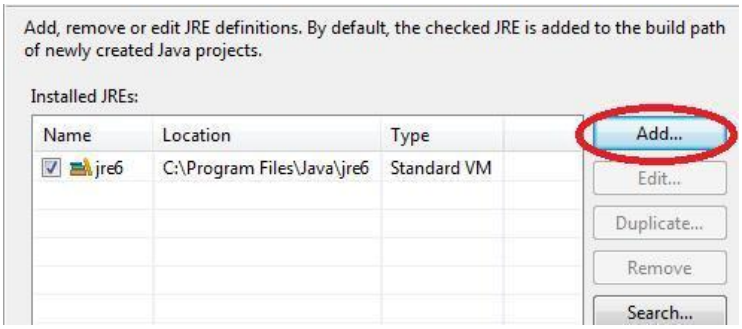
In the above scenario previous version was 1.6 which is now changed to 1.5

2.2 Installed JREs

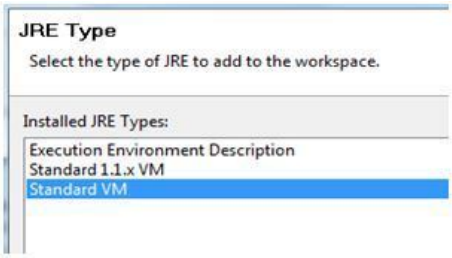
You can also add/remove JRE definitions using Installed JREs from Java of Preferences tab.



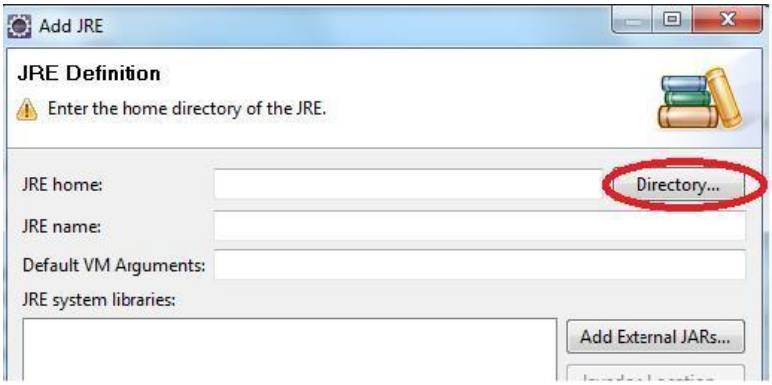
Now the default JRE is already checked in the Installed JREs, you can add new JRE using Add button and to remove, select the existing JRE and click Remove button.



On clicking Add a window is opened where you need to select JRE Type

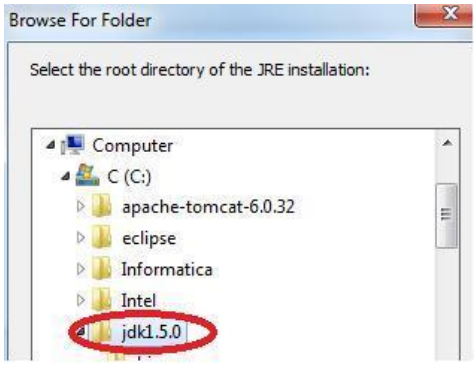


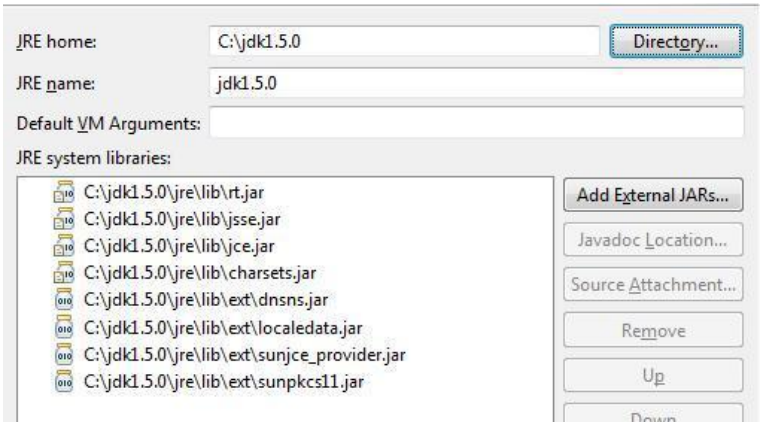
Select the type of JRE and click Next. Now you need to add JRE definition, for this you need to click on Directory.



Choose the appropriate folder from the respective location and click OK

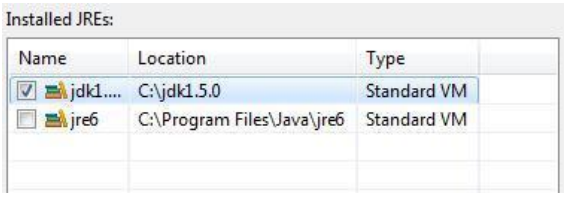
Now the library files related to the new definition are added. Click Finish to add it.





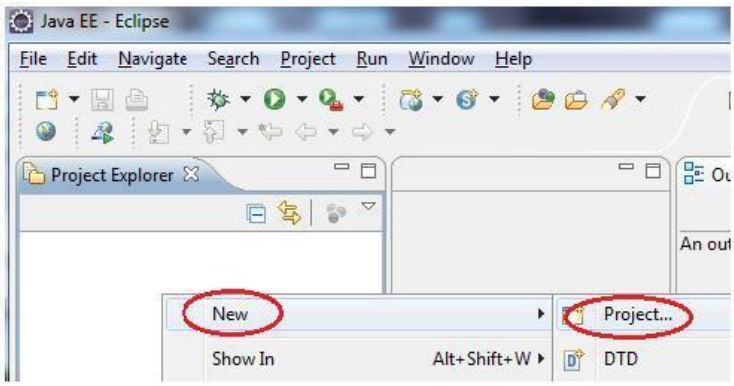
Select default JRE and click OK.

Now the setup is done.

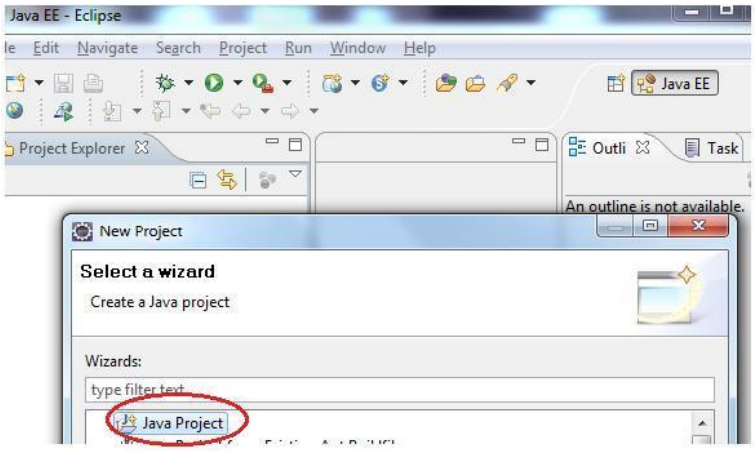


3. Create Java Project

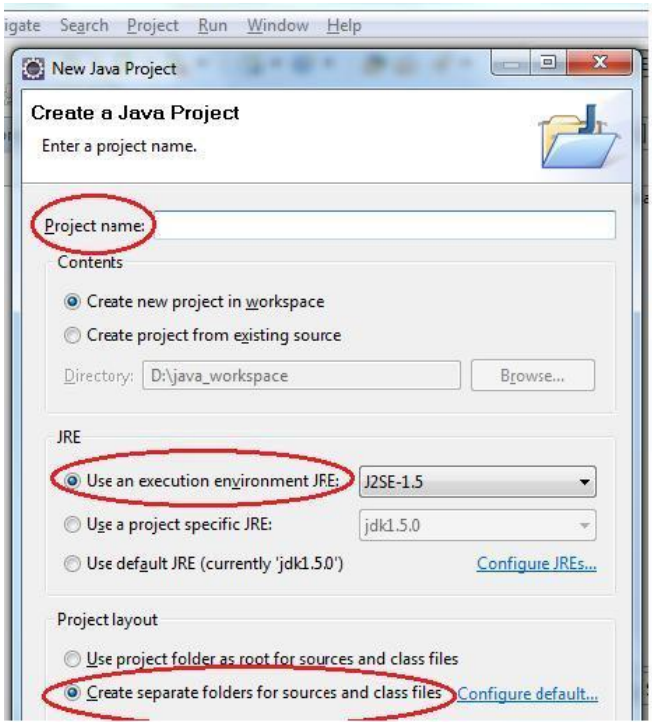
Right click anywhere in the Project Explorer area. Select New → Project.



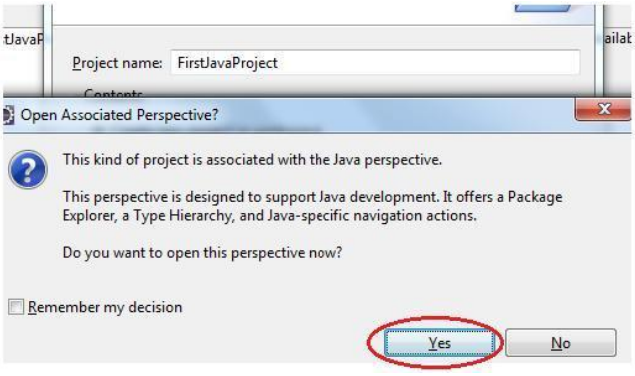
A wizard to create new project will appear. Select Java Project from the list of options available. Click Next.



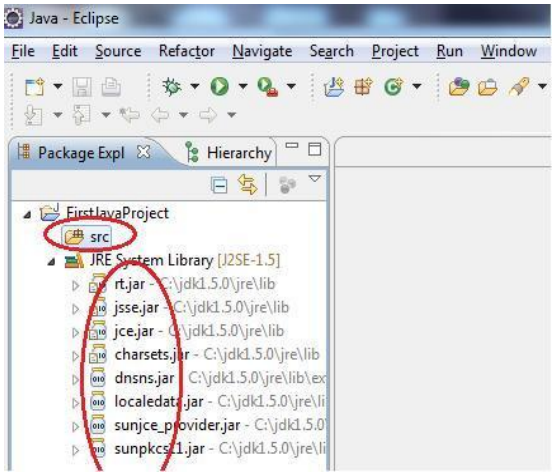
It opens a wizard to configure few setting for the new project. Enter the project name of your choice – eg. FirstJavaProject. Make sure your have select the below highlighted options. Click Finish.



Eclipse may ask you to open the associate perspective. Click Yes.

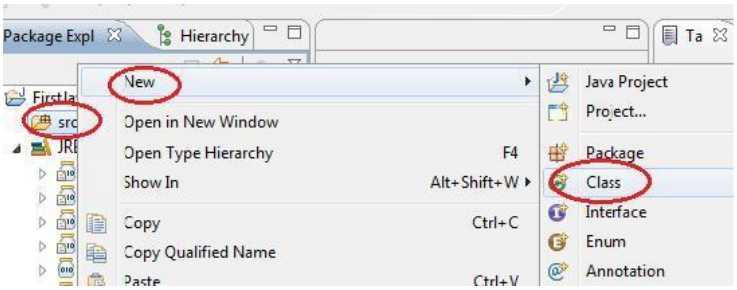


4. Project Directory.



5. Create Java Class

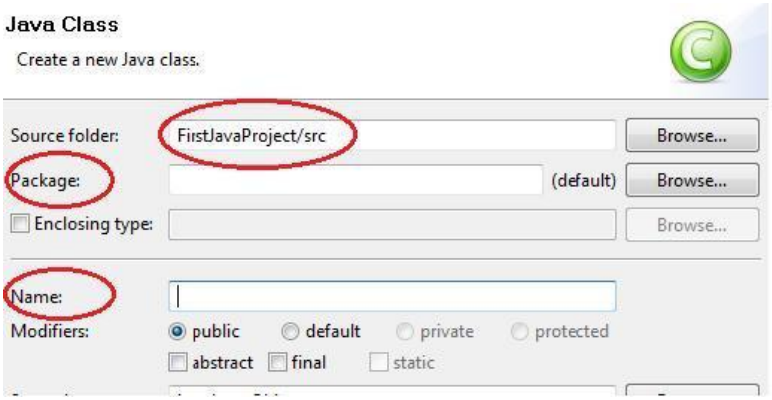
Right click on src folder. Select New → Class



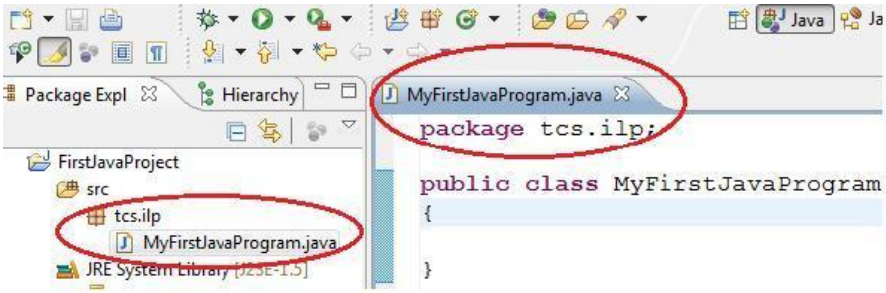
Reconfirm the location where you want the class to be created – its is in the src folder.

Enter the package name – **tcs.ilp**

Enter the class name - **MyFirstJavaProgram**

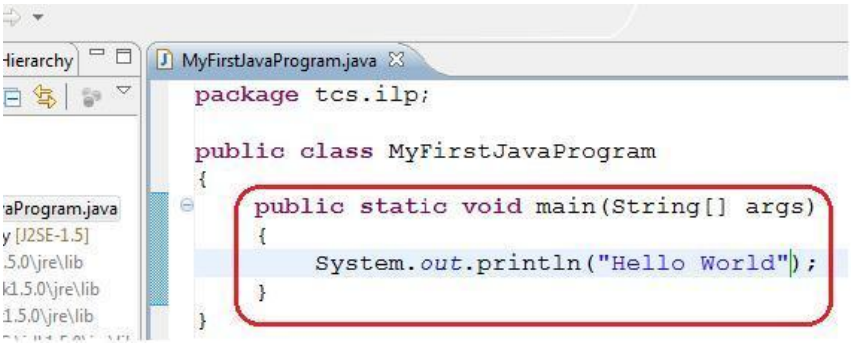


Eclipse created a package tcs.ilp under the src folder. It also creates a java file MyFristJavaProgram.java. Double click on the MyFristJavaProgram.java and eclipse opens the contents of the file in a window.

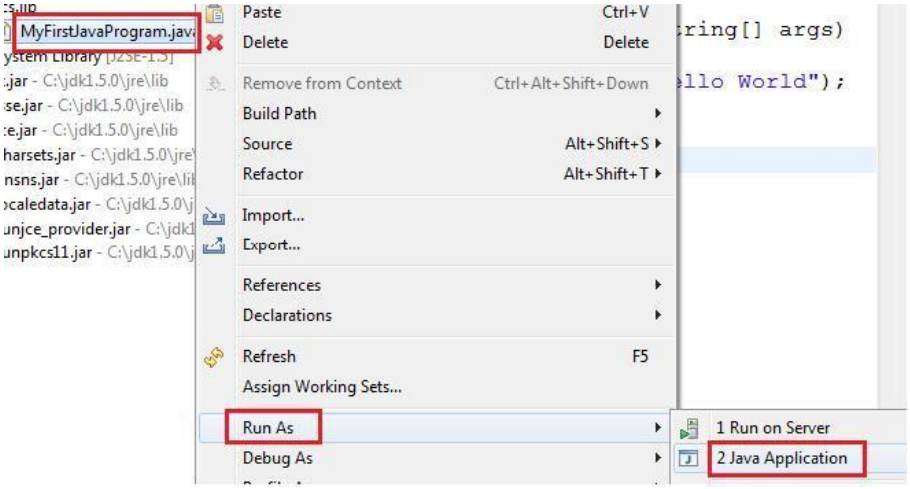


Add the main method and a simple print statement.

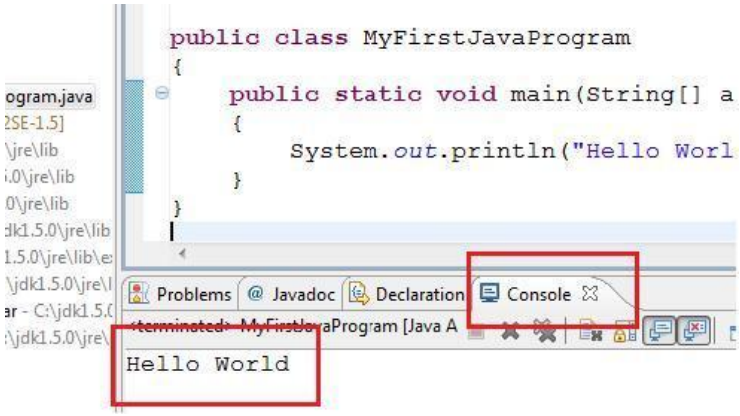
6. Execute the Java program



Right click on the file MyFirstJavaProgram. Select Run As → Java Application.



Here you do not need to compile the java file as eclipse does it for us automatically. We just need to run the program. Eclipse provides a console window where you can see the output of the program that you executed as Java Application.



Ask a doubt *(Misuse of 'Ask a Doubt' Section will be dealt as per the Terms & Conditions of Campus Commune)*

Open Doubts | **Closed Doubts**

SASIVINEETHA GATTUPALLE

What does the following statements mean?

1.Use an execution environment JRE

2.Use a project specific JRE

about 7 days ago

SASIVINEETHA GATTUPALLE

I have installed jdk 1.8.0_11. Now which eclipse package should I install inorder to run java programs? Please help me. I'm confused.... Thanks in advance.

about 7 days ago

SAILESH SEERA

windows 64bit it was not opening

about 8 days ago

MEGHA GOYAL

I am unable to find Eclipse Galileo for Windows 64 bit... plz provide link for it...

about 15 days ago

VISHWA DEEPAK

I think Eclipse Galileo is not available for windows 64 bit. is it ? if yes please provide the link for download .

about 18 days ago

ATUL VISWANATH

Download Eclipse Luna ver 4.4 64-bit instead!

ABHISHEK SAH

What is the difference in using Eclipse Galileo, eclipse juno or myeclipse ?

about 20 days ago

SOURAV GHOSH

Eclipse Galileo and Eclipse Juno represent codenames for two different release versions of Eclipse. Instead of using the traditional version numbers directly for branding, each simultaneous release versions are given code names. Presently, the versions are named using words whose first letter increments by one with each successive release version. For example, Eclipse **Galileo** is followed by Eclipse **Helios, Indigo, Juno, Kepler**, and the latest version to-date, Eclipse **Luna**.

SRIJAN GUPTA

in the given link eclipse galileo for 32 bit windows system can be downloaded.. will it run in 64 bit system also??

about 21 days ago

RAJENDRA LAMROR

no

download for 64 bit

SOURAV GHOSH

Yes, it will run. A 64 bit OS can run both 64 bit software as well as 32 bit software. However, 32 bit OS can only run 32 bit software. So, if you have 64 bit OS running on your system, you can download either 32 bit eclipse or 64 bit eclipse. Both will run perfectly on your system. However, I'd recommend that you install the same bit length JDK on your machine, i.e. if you install 32-bit version of eclipse you should link it with a 32-bit version of JDK, and vice versa.

RIZWANA AHMAD

hey...

i downloaded java from the link provided but i somehow ended up at jdr 1.7 n eclipse dont support that, can anyone plz give me the correct link of jdk n jre for windows 8.1 64bit...!! im stucked at this point from 2 days..plz help :(

about 23 days ago

MAYUR GUPTA

install jdk 5

MAYUR GUPTA

Link to install jdk 5 :

http://www.oracle.com/technetwork/java/javasebusiness/downloads/java-archive-downloads-javase5-419410.html

Java Lounge

☐Logout

Java

1.IT Application Overview And Features	Q
2.Need of Programming and Introduction to OOP approach	Q
3.Introduction to Java,JVM,JDK	Q
4.Operators and Variables in Java	Q
5.Introduction to class in java,access specifiers, this and static in Java	Q
6.Java Library, Packages, Use of import	Q
7.Conditional operations	Q
8.Basic Iterations and Arrays	Q
9.Designing Web Pages Part 1(HTML and CSS)	Q
10.Designing Web Pages Part 2 (JavaScript and JQuery)	Q

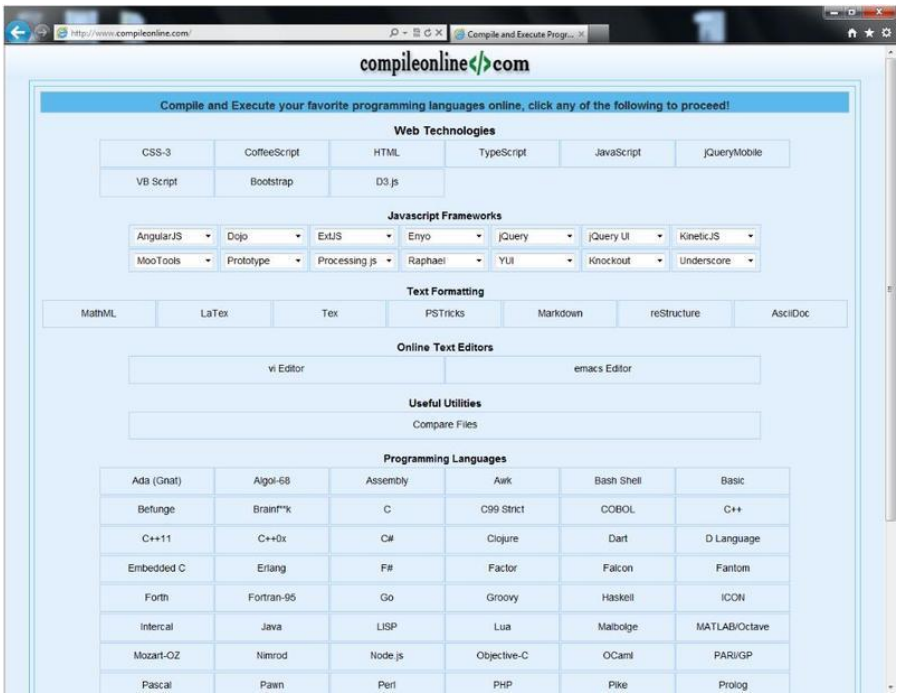
☐ Course Completion Quiz

3.8. Compileonline.com

Compileonline.com is a web application which is used to compile and execute programming languages online. You can compile java programs as well. Here is the quick guide to compile and execute your program.

Step1:

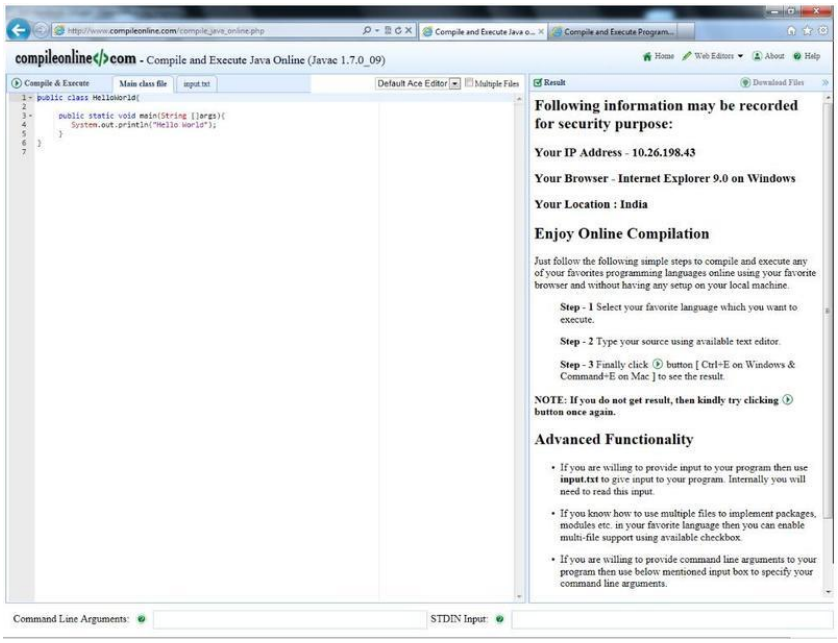
Open browser and type www.compileonline.com in the address bar.



Step 2:

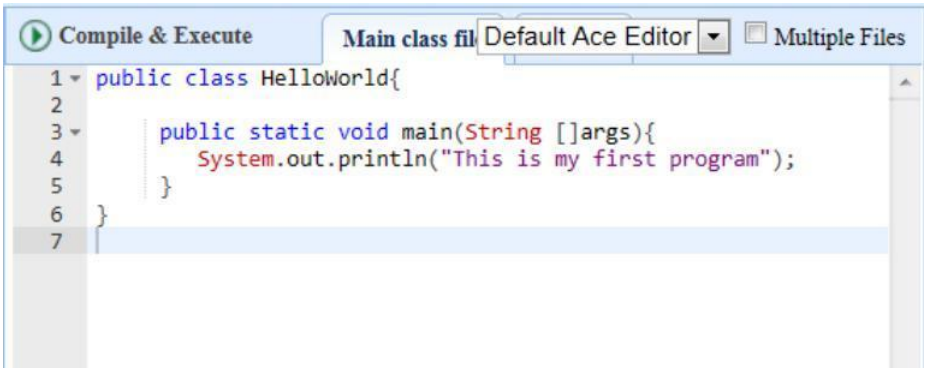
Click on "Java" in the Programming Languages section.

Now a window appears something like this:



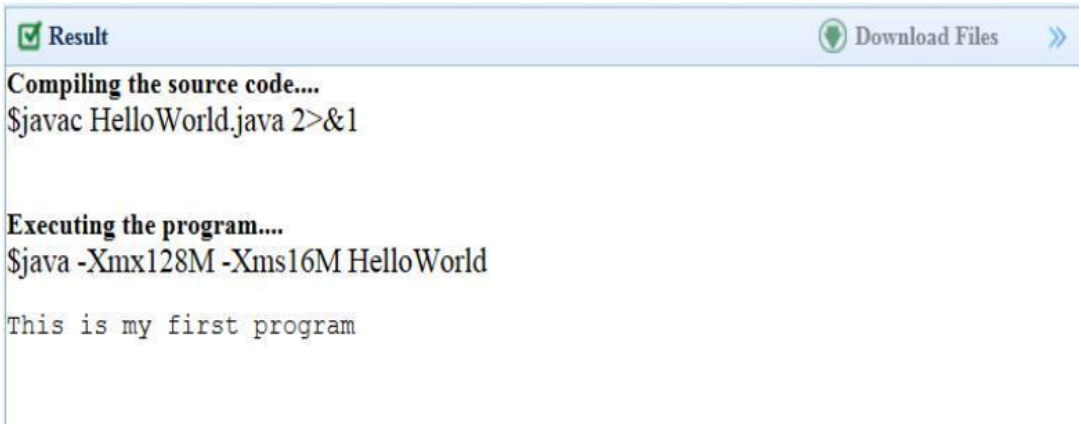
The left hand area is where the source code is written and the right side "Result" tab is where the output is displayed. Steps to compile and execute the programs are given on the right hand area. We will now execute a sample code.

Step 3: Write a program on the left hand area in "Main class file" tab



Step 4: Finally click the following button [Ctrl+E on Windows & Command+E on Mac] to see the result.

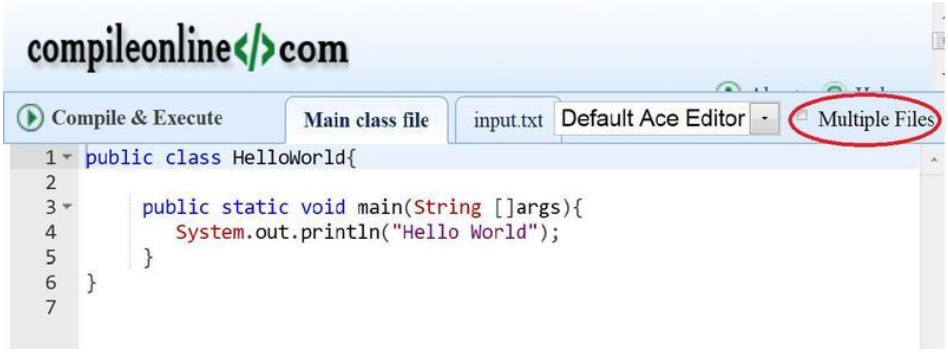
Step 5: The result is displayed on the right hand side



That's it!! Done!!

This application also helps us to execute multiple class files let us also see a small demo on how to execute multiple class files using compile online.

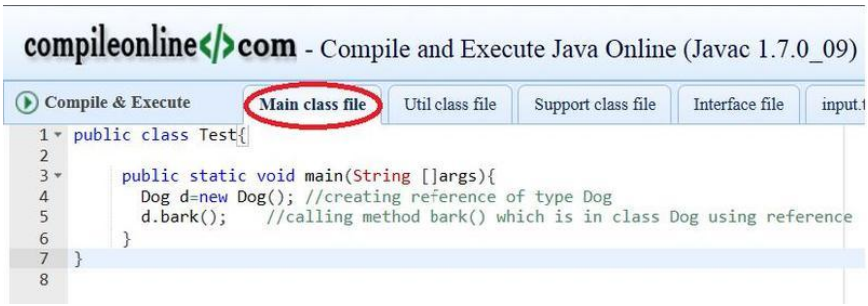
Step 1: Check the check box named "Multiple Files"



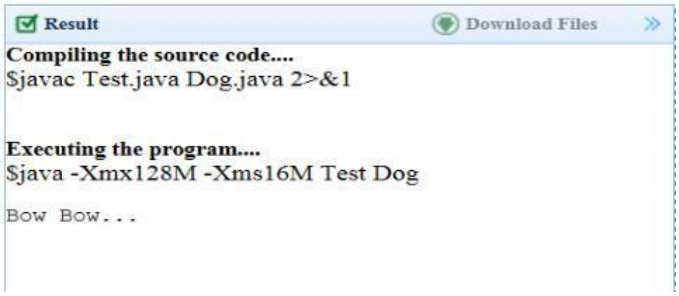
Step 2: Click "Support class file" and write the program in the area provided

In the above program, we have created a Dog class which has a method bark(). Now to execute the above program, we need a Test class which contains main() method and a call to bark() method.

Step 3: Click Main class file tab and create a class with main() method



Step 4: Click Compile & Execute.



Ask a doubt *(Misuse of 'Ask a Doubt' Section will be dealt as per the Terms & Conditions of Campus Commune)*

3.9. Reference Links & videos

Reference links

http://www.devmanuals.com/tutorials/java/corejava/JavaHistory.html

http://www.freejavaguide.com/history.html

http://www.devmanuals.com/tutorials/java/corejava/IntroductionJava.html

http://www.w3resource.com/java-tutorial/download-and-Install-JDK-Eclipse-IDE.php

http://docs.oracle.com/javase/tutorial/getStarted/application/index.html

http://www.devmanuals.com/tutorials/java/corejava/JavaEclipseHelloWorld.html

http://www.devmanuals.com/tutorials/java/corejava/JavaVirtualMachine.html

http://viralpatel.net/blogs/java-virtual-machine-an-inside-story/

http://www.javatpoint.com/internal-details-of-jvm#jvminternalarch

http://docs.oracle.com/javase/tutorial/getStarted/intro/definition.html

http://en.wikipedia.org/wiki/Java_Development_Kit

http://www.javatpoint.com/difference-between-jdk-jre-and-jvm

Ask a doubt *(Misuse of 'Ask a Doubt' Section will be dealt as per the Terms & Conditions of Campus Commune)*

Open Doubts | **Closed Doubts**

ANUGRAH TIWARI

the quiz is not opening for this module.

kindly respond quickly

thanking you in advance

about 2 months ago

TATA CONSULTANCY SERVICES

Java Lounge

Logout

Java

Course Completion Quiz

4.Operators and Variables in Java

Objective

- Understand Java Variables
- Understand numeric data types, basic operations and expressions and operator precedence.
- Perform simple and complex numeric computations
- Ability to choose right numeric data types and order of computation
- Understanding this keyword

4.4 Operators

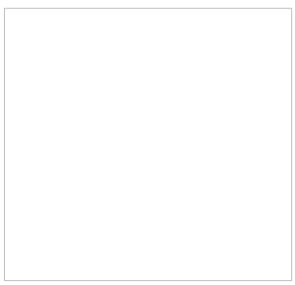
4.5 String

4.6 Reference Links & videos

4.7 Practice Problems

4.1. Java Variables

- In Java, the attributes in classes are called as Variables.
- A variable is a container that holds values that are used in a Java program.
- Variable is declared with a type of data it stores.
- All variables should be declared before they are used in the program.
- As shown below, variables have a name, value and type.



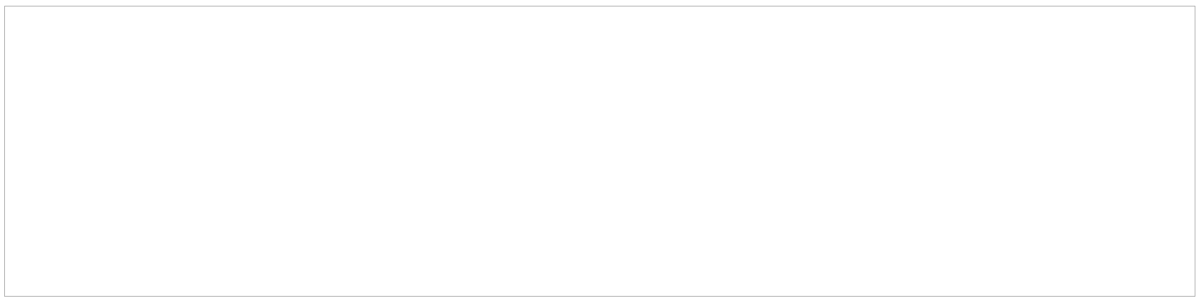
Variable declaration

The basic form of a variable declaration is shown:

<data type> <variable name> = <initial value>

Example:

int age =20;
Here age is a variable of type integer with initial value of 20.



Note: Product.java example which is been given that can not be executed as main method is not there.

Types of Variables

Member / Instance variables:

Instance variables are declared in a class, but outside a method, constructor or any block.

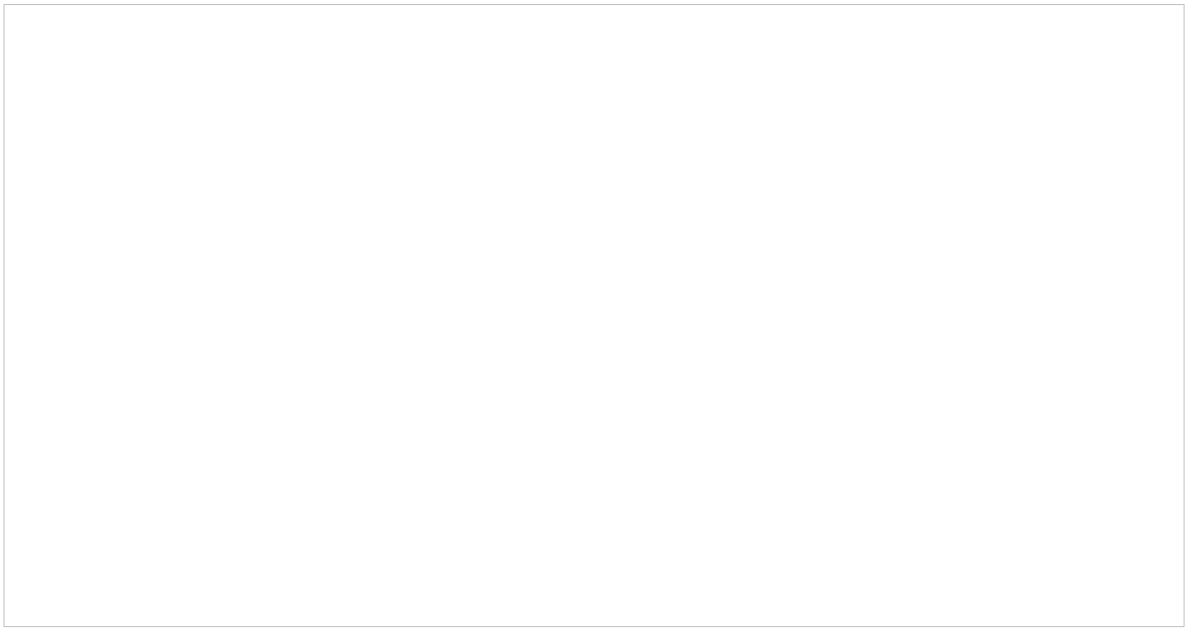
Access modifiers such as private, public, protected can be specified for instance variables.
They are visible for all methods, constructors and blocks in the class.
Values can be assigned during the declaration or within the constructor.

Example 1:
In this example we can see how to declare and assign initial value to the variables. We can also see the access modifiers used with the variable declaration. Here we have used private access modifier.

Note: Product.java example which is been given that can not be executed as main method is not there.

Example 2:

This example demonstrates the default values of instance variables.



Note:Employee.java example which is been given that can not be executed as main method is not there.

The default initial values of the data members of Employee class are as shown below:

name is null (as it is a String object reference)

id is 0(as it is primitive type : integer)

basicSalary is 0.0 (as it is primitive type : float)

Local variables

- Local variables are declared inside methods, constructors or blocks.
- Access modifiers cannot be used for local variables.
- They are visible only within the declared method, constructor or block.
- There is no default value for local variables. So local variables should be declared and an initialized before its first use.

Example:

This example demonstrates how to declare and use local variables.

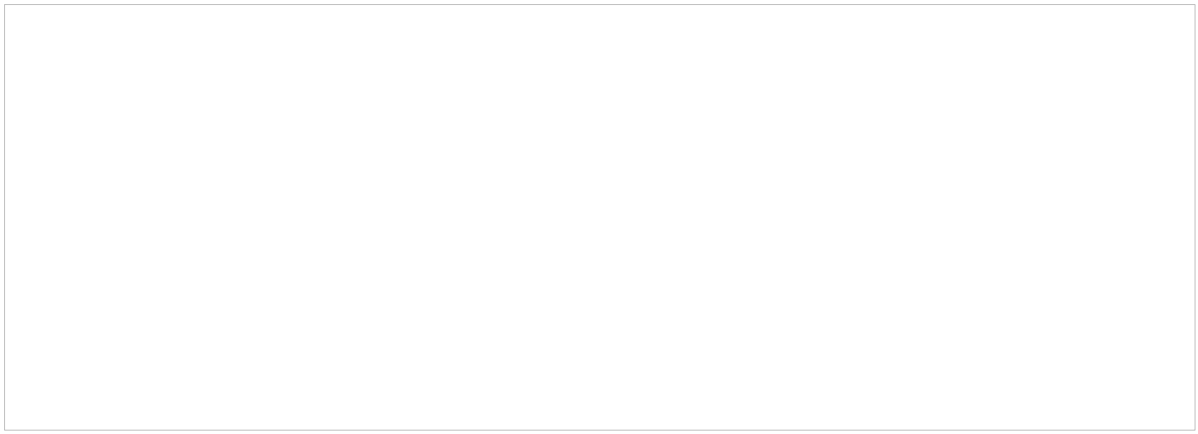
Note: Calculator.java example which is been given that can not be executed as main method is not there.

Here "d" is the local variable created inside the calculation method.

Parameters

Parameters are the list of arguments that come into the function. Parameters are used to pass values to functions and to receive values in a function.

Example: This example demonstrates to how to use parameters to receive values.

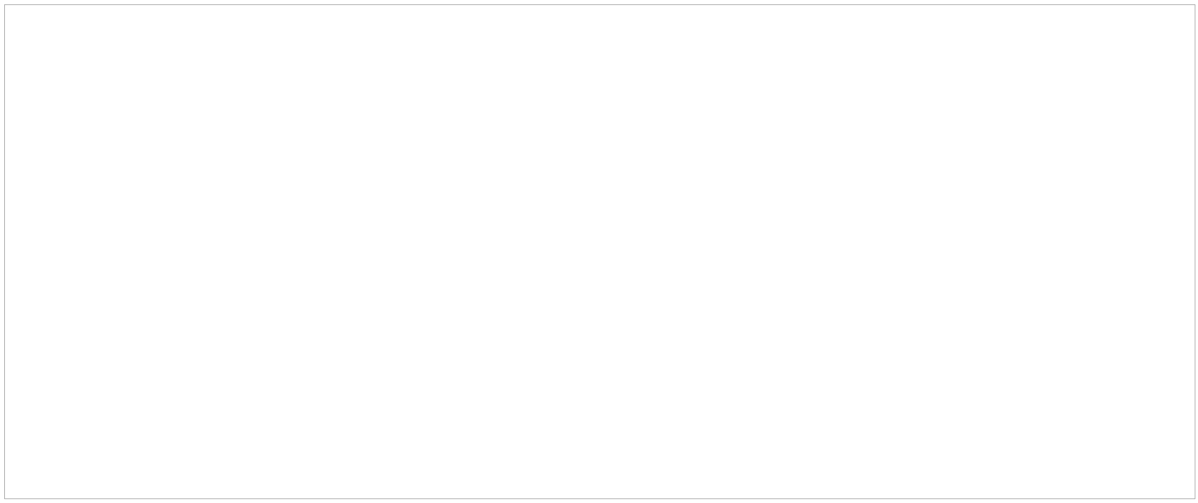


Here we are using **int a, int b** as parameters to receive values sent to add method.

Note:Calculator.java example which is been given that can not be executed as main method is not there.

Example 2:

This example demonstrates to how to use parameters to send values to functions.



In c.add(a,b) a and b act as parameters, which are used to send values to add method.

Note: Until, you will not include import statement or if both programs (Tester.java, Calculator.java) are not in same package that program will not run.

Ask a doubt *(Misuse of 'Ask a Doubt' Section will be dealt as per the Terms & Conditions of Campus Commune)*

Open Doubts | **Closed Doubts**

ORUGUNTA MEGHANA

i am not able to see figure examples for this topic only,,is this happening to anyone else also??

about 1 month ago

VISHNU PRAKASH

Try changing the security settings in your browser.

KRISHNA DHARMARAJ

ya I am also facing the same problem

POORNIMA MURUGESAN

Try in chrome. I had the same problem in mozilla.

NANDHINI R

I too had the same problem..open the image in a new window and you can view it..It worked for me..

BALAJI PATRO

wat settings do we hav to change?

KALYAN PILLA

The CA certificate provided by campus commune is not identified by most browsers.

If you ever had to "Add Exception" to SSL certificate provided by nextstep, this can be your problem.

Solution:

1. Right click on the place holder for the image
2. Select Open Image in a new tab
3. Add Exception when a Security Warning appears (Now the images, *probably present on a different server*, are allowed)
4. Continue browsing the course in a usual manner. Now, the images will load.

This worked for me with Chrome on Trisquel.

PRIYANKA MUKHERJEE

Open the image in a new window.

PAYAL DASH

i hv nt yet cmpleted my task but it shows dat u r failed n my 2nd chance also faited,....how can it be????

about 1 month ago

ABDUL HAQ

Can any one please tell me what are the access modifiers??

about 2 months ago

VIDHI MITTAL

Access modifiers basically mean the way in which each variable or method can be accessed. java supports public, protected and private access modifiers.

ABDUL HAQ

Thanks Mittal ji

LAKSHMINADHA PRASAD THOTA

In java there are four Access Modifiers are available

1)public String name;// public is accessible to any where or any package

2)private String name;// private is accessible only with in the class

3)protected String name; // protected is accessible with in the class or with in the same package and the classes which are inherited even though they are in different packages

4)String name; // this is the **default** one we can not specify any thing.is accessible with in the class or with in the same package

the default can access in the same class and other classes which are in same package. it is also called**'package-private' or 'default' or 'friendly' access? modifier.**

4.2. Static Variables and Functions

Static Variables

i. Class variables also known as static variables, are declared with the static keyword in a class.
ii. These are declared outside a method or constructor.

Eg: public static int count = 0;

iii. The basic concept behind static is, there would only be one copy of each class variable per class, regardless of how many objects are created from it.
iv. That means, for a static field, regardless of the number of instances created, will hold the same data across all objects.
v. If the value in static variable is changed, it is reflected in all objects. On the other hand, non-static fields of objects store their individual state. The value of a non-static field (also called instance variables) is unique for every instance / object.

vi. Static variables are created when the program starts and destroyed when the program stops.
vii. Default values are same as instance variables. For numbers, the default value is 0; for boolean, it is false; and for object references, it is null.
viii. Values can be assigned during the declaration or within the constructor.

ix. Static variables in a class can be accessed from a second class by calling with the class name of the first. ClassName.variableName. Thus we can see that the static variables are not called by objects. It is directly called with class name.

Example 1:

Following example demonstrates how to declare and use static variables.

StaticExample.java

```
class StaticExample {

    public static int num;      //Here num is a static variable.

    public void add(int n, int m){
         num = n + m;
        System.out.println("Result ="+num);
    }

}
```

Note: StaticExample.java example which is been given that can not be executed as main method is not there.

StaticExample.java TestStatic.java

```
public class TestStatic {


    public static void main(String[] args) {

        System.out.println("Value of num="+StaticExample.num); // Use of static variable.

        StaticExample s = new StaticExample();
        s.add(10,15);

        System.out.println("Value of num="+StaticExample.num); // Use of static variable.

    }

}
```

Console

```
<terminated> TestStatic [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (Mar 28, 2014 9:30:00 AM)
Value of num=0
Result =25
Value of num=25
```

In StaticExample class we have declared a static variable called num.

In TestStatic class we are accessing that variable.

We have created object of StaticExample class in TestStatic class to access the attributes and methods of StaticExample class.

Since num is a **static** variable, we can access it directly with the **class** name, before creating object of that **class**.

Static Functions

Through these courses we have already come across a static function. Can you guess? Yes, it's nothing but the main method which we used to test the classes and functions which we write.

Static methods are conceptually the same as static variables, thus the reasons to use or not use them are similar. They belong to the class, not specific objects of that class. There is no need of a logical instance variable to call these methods. That is, if a method needs to be in a class, but not tied to an object, then it make it static. If the method is more logically part of an object, then it should not be static.

Main is static because JVM can call main without creating an object first. It probably simplified the design of the JVM.

Static methods are also accessed by calling the method with the class name. ClassName.methodname.

Example: This example demonstrates use of static methods.

Calculator.java

```
public class Calculator {

    public static int add(int num1, int num2){  // Here add is a static method
        int result = num1 + num2;
        return result;

        }
}
```

Note: Calculator.java example which is been given that can not be executed as main method is not there.

Calculator.java | TestCalculator.java

```
public class TestCalculator {


    public static void main(String[] args) {
        int total = Calculator.add(10,12);
        System.out.println("Total ="+total);

        total = Calculator.add(30,10);
        System.out.println("Total ="+total);

    }

}
```

Console

```
<terminated> Test [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (Mar 28, 2014 8:52:34 AM)
Total =22
Total =40
```

Here add is a static method written in calculator class. We are accessing that in TestCalculator class.
Since it is static method we can access it without creating object of that class, directly with the classname.

Points to remember

1. Static methods can't use non-static, instance variables directly.
2. Static methods can't use non-static, methods directly.

That means, if a static method need to access a non-static variable or method, then first create an object and using that object you need to access the method or variable.
This is because; static members which are related directly to class cannot see instance variable state and behaviour.

Ask a doubt *(Misuse of 'Ask a Doubt' Section will be dealt as per the Terms & Conditions of Campus Commune)*

4.3. Numeric Data types

Numbers are so important in Java that 6 of the 8 primitive data types are numeric types which are shown in tabular format below.

There are both integer and floating point primitive types. Integer types have no fractional part whereas floating point types have a fractional part.

Description of each type is given after the table.

Type	Size	Range	Example
byte	8 bits	-128 to +127	byte b = 120;
short	16 bits	-32768 to +32767	short s = 1200;
int	32 bits	-2,147,483,648 to +2,147,483,647	int num = 120000;
long	64 bits	-9,223,372,036,854,775,808 to +9,223,372,036,854,775,807	long d = 12345678998 l;
float	32 bits	1.40129846432481707e-45 to 3.40282346638528860e+38 (positive or negative).	float f = 123.45f;
double	64 bits	4.94065645841246544e-324d to 1.79769313486231570e+308d (positive or negative).	double d = 12345.7856;

Each of these types uses a fixed number of bits. i.e. same number of bits will be used no matter what value is represented.

For example, all values represented using the short data type use 16 bits. The value zero (as a short) uses 16 bits and the value thirty thousand also uses 16 bits.

1] byte:

- byte data type is an 8-bit signed two's complement integer.
- Minimum value is -128 (-2^7)
- Maximum value is 127 (inclusive)(2^7 -1)
- Default value is 0
- byte data type is used to save space in large arrays, mainly in place of integers, since a byte is four times smaller than an int.
- Example: byte a = 100 , byte b = -50

2] short:

- short data type is a 16-bit signed two's complement integer.
- Minimum value is -32,768 (-2^15)
- Maximum value is 32,767 (inclusive) (2^15 -1)
- short data type can also be used to save memory as byte data type.
- A short is 2 times smaller than an int.
- Default value is 0.
- Example: short s = 10000, short r = -20000

3] int:

- int data type is a 32-bit signed two's complement integer.
- Minimum value is - 2,147,483,648.(-2^31)
- Maximum value is 2,147,483,647(inclusive).(2^31 -1)
- int is generally used as the default data type for integral values unless there is a concern about memory.
- The default value is 0.
- Example: int a = 100000, int b = -200000

4] long:

- long data type is a 64 bit signed two's complement integer.
- Minimum value is -9,223,372,036,854,775,808.(-2^63)
- Maximum value is 9,223,372,036,854,775,807 (inclusive). (2^63 -1)
- This type is used when a wider range than int is needed.
- Default value is 0L.
- Example: long a = 100000L, int b = -200000L

5] float:

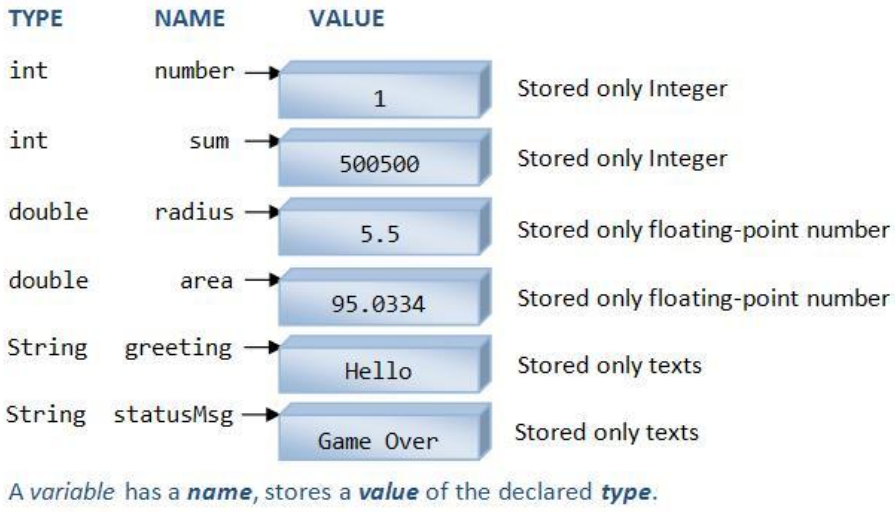
- float data type is a single-precision 32-bit IEEE 754 floating point.
- float is mainly used to save memory in large arrays of floating point numbers.
- Default value is 0.0
- float data type is never used for precise values such as currency.
- Example: float f1 = 234.5f

6] double:

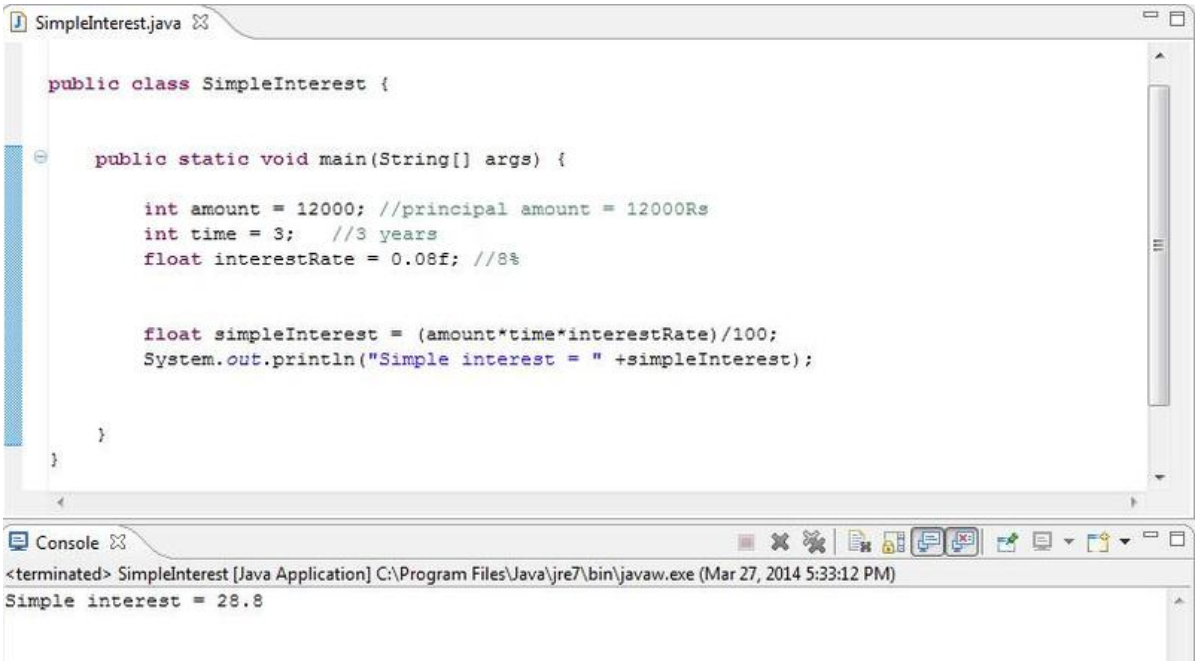
- double data type is a double-precision 64-bit IEEE 754 floating point.

- This data type is generally used as the default data type for decimal values, generally the default choice.
- double data type should never be used for precise values such as currency.
- Default value is 0.0d.
- Example: double d1 = 123.4

Fig: This digram demonstrates use of different datatypes.



Example : This program demonstrate the use of data types.



The result has fractional value, so we are using float variable to store it

Ask a doubt *(Misuse of 'Ask a Doubt' Section will be dealt as per the Terms & Conditions of Campus Commune)*

Open Doubts | **Closed Doubts**

SAYANTAN GANGOPADHYAY

how we initialize bytes????...as we know byte a=20;...but they declared it as a wrong answer!!!!!!!!!!

about 13 days ago

SAYANTAN GANGOPADHYAY

stream of data is fetched from network or file by using which data type????

about 13 days ago

LAKSHMI D

char

RIZWANA AHMAD

which data type should be used to store currency?? if not double..?is there anything like long double in java? or any toher data type that is specially used for this purpose

about 22 days ago

ABHISHEK SAH

Currency is a highly precise value so Java provides a class named "**Currency**" itself for it. The data type suitable is BigDecimal.
For more refer: http://docs.oracle.com/javase/7/docs/api/java/math/BigDecimal.html

SIDDHARTH GOSALIA

In the example program given above, instead of storing the answer as 'float', can I store it as 'double'. What difference will it make to the program as both of them can store decimal values?

about 26 days ago

SHWETA NAYAK

yes you can store it as double. but the memory required for storing a value is more, if double is used since its size is 64 bits

4.4. Operators

Java provides a rich set of operators to manipulate variables. We can divide all the Java operators into the following groups:

- Arithmetic Operators
- Relational Operators
- Logical Operators
- Assignment Operators
- Misc Operators

1] **The Arithmetic Operators:**

Arithmetic operators are used in mathematical expressions in the same way that they are used in algebra. The following table lists the arithmetic operators:

Assume integer variable A holds 10 and variable B holds 20, then:

Operator	Description	Example
+	Addition – Adds values on either side of the operator	A + B = 30
-	Subtraction – Subtracts right hand operand from left hand operand	A - B = -10
*	Multiplication – Multiplies values on either side of the operator	A * B = 200
/	Division – Divides left hand operand by right hand operand	B / A will give 2
%	Modulus – Divides left hand operand by right hand operand and returns remainder	B % A will give 0
++	Increment – Increases the value of operand by 1	B++ will give 21
--	Decrement – Decreases the value of operand by 1	B-- will give 19

2] The Relational Operators:

There are following relational operators supported by Java language

Assume variable A holds 10 and variable B holds 20, then:

Operator	Description	Example
==	Checks if the values of two operands are equal or not. If equal then condition becomes true.	A == B is not true
!=	Checks if the values of two operands are equal or not. If not equal then condition becomes true.	A != B is true
>	Checks if the value of left operand is greater than right operand. If yes then condition becomes true.	A > B Is not true
>=	Checks if the value of left operand is greater than or equal to right operand. If yes then condition becomes true.	A >= B Is not true
<	Checks if the value of left operand is smaller than right operand. If yes then condition becomes true.	A < B is true
<=	Checks if the value of left operand is smaller than or equal to right operand. If yes then condition becomes true.	A<=B is true

3] The Logical Operators:

- Logical operators return a true or false value based on the state of the Variables.
- There are six logical, or boolean, operators. They are AND, conditional AND, OR, conditional OR, exclusive OR, and NOT.
- Each argument to a logical operator must be a boolean data type, and the result is always a boolean data type.
- An example program is shown below that demonstrates the different Logical operators in Java.

Example:

In this example you can see the use of logical operators.

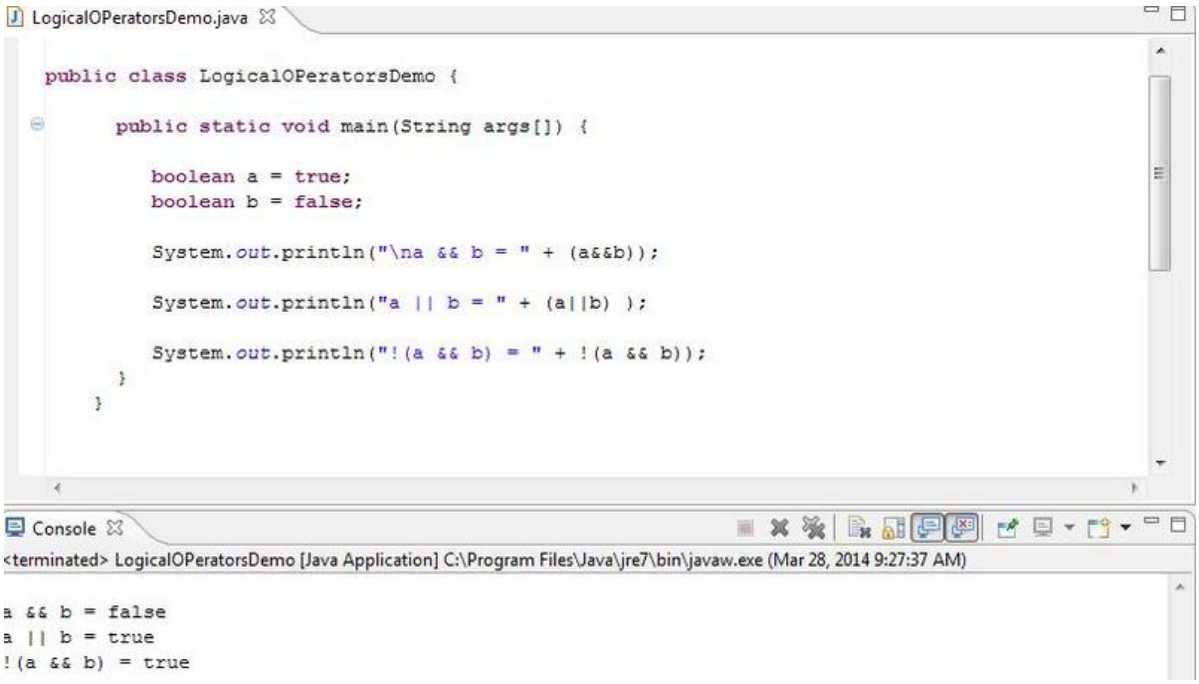


Fig: Logical operators

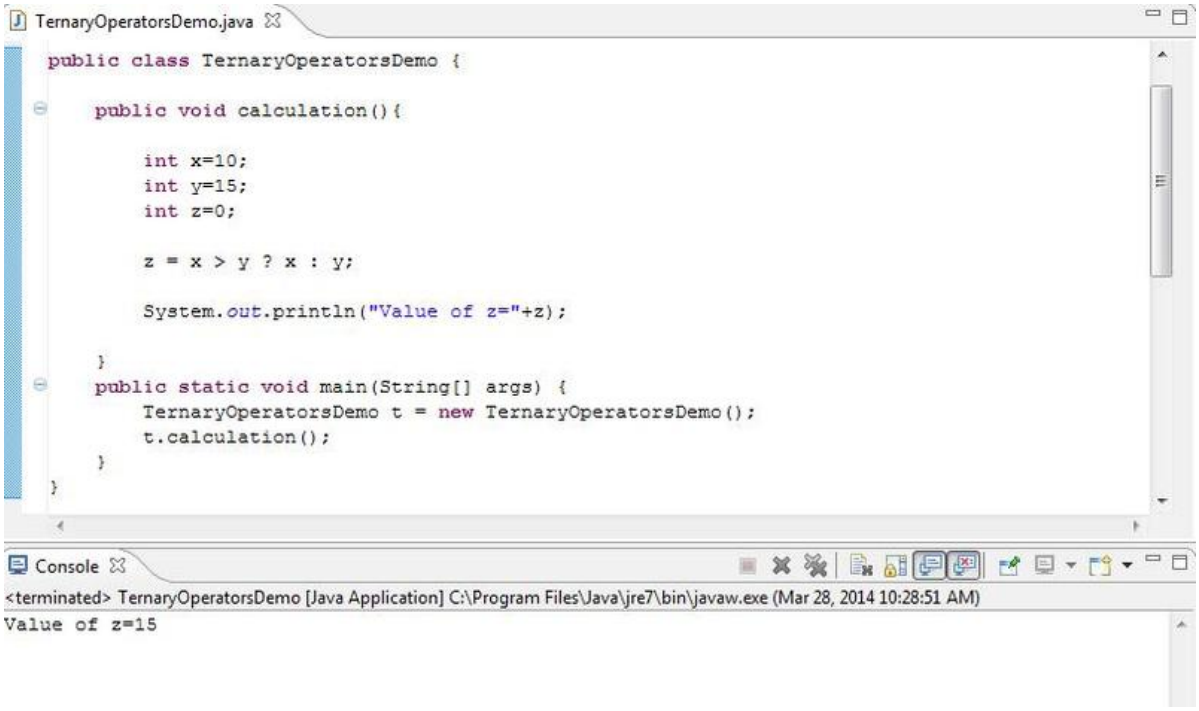
x	y	!x	!y	x&y x&&y	x | y x || y	x ^ y
TRUE	TRUE	FALSE	FALSE	TRUE	TRUE	FALSE
TRUE	FALSE	FALSE	TRUE	FALSE	TRUE	TRUE
FALSE	TRUE	TRUE	FALSE	FALSE	TRUE	TRUE
FALSE	FALSE	TRUE	TRUE	FALSE	FALSE	FALSE

5] The Assignment Operator:

- Assignment operator is used to assign values to variables.
- " = " is used as assignment operator. Eg: int i=10;
- The variable are always on the left-hand side of the assignment operator and the value to be assigned is always on the right-hand side of the assignment operator.
- The assignment operator is evaluated from right to left, so a = b = c = 0; would assign 0 to c, then c to b then b to a.
- i = i + 2; // same as i=10+2
- Here we say that we are assigning i's value to the new value which is i+2.
- A short cut way to write assignments like this is to use the += operator. It's one operator symbol so don't put blanks between the + and =. So it is also called as short hand operator.
- i += 2; // Same as "i = i + 2

6] The Conditional Operator or Misc operator:

- The Conditional operator is the only ternary (operator takes three arguments) operator in Java. The operator evaluates the first argument and, if true, evaluates the second argument. If the first argument evaluates to false, then the third argument is evaluated.
- The conditional operator is the expression equivalent of the if-else statement.
- An example program is shown below that demonstrates the Ternary operator in Java.
- This example demonstrates how conditional operator can be used to find grater of 2 numbers.



Operators can also be classified as follows:

Unary Operators

The unary operators require only one operand. They work with single operands.

Example:

count ++; // Existing value in variable count is incremented by 1.

count --; // Existing value in variable count is decremented by 1.

Binary Operators

Binary operators work with two operands or variables.

Example:

int age = 35; // value 35 assigned to variable age.

int x = 5; int y = 3;

int z = x + y; // adding values of two variables and assigning result to another variable.

Ternary Operators

In ternary operator, takes 3 components: condition, true result, false result. This operator is having a ? and :. It looks like result = condition ? value 1 : value 2

Example:

int a = 10;

int x = a > 0 ? 1: 0; // checking a > 0, if yes returns 1, if no returns 0.

Precedence of Operators

Java has well-defined rules for specifying the order in which the operators in an expression are evaluated when the expression has several operators. For example, multiplication and division have a higher precedence than addition and subtraction. Precedence rules can be overridden by explicit parentheses.

Precedence Order:

When two operators share an operand the operator with the higher precedence goes first. For example, 1 + 2 * 3 is treated as 1 + (2 * 3), whereas 1 * 2 + 3 is treated as (1 * 2) + 3 since multiplication has a higher precedence than addition.

Associativity:

When two operators with the same precedence the expression is evaluated according to its associativity. For example x = y = z = 17 is treated as x = (y = (z = 17)), leaving all three variables with the value 17, since the = operator has right-to-left associativity (and an assignment statement evaluates to the value on the right hand side). On the other hand, 72 / 2 / 3 is treated as (72 / 2) / 3 since the / operator has left-to-right associativity.

Precedence and associativity of Java operators:

The table below shows all Java operators from highest to lowest precedence, along with their associativity. Here, operators with the highest precedence appear at the top of the table, those with the lowest appear at the bottom. Within an expression, higher precedence operators will be evaluated first.

For more information on precedence and associativity you can visit the following link.
http://docs.oracle.com/javase/tutorial/java/nutsandbolts/operators.html

Ask a doubt *(Misuse of 'Ask a Doubt' Section will be dealt as per the Terms & Conditions of Campus Commune)*

4.5. String

Strings are an incredibly common type of data in computers. Strings, which are widely used in Java programming, are a sequence of characters. In the Java programming language, strings are objects.

Java string is a series of characters gathered together, like the word "Hello", or the phrase "practice makes perfect". Create a string in the code by writing its chars out between double quotes.

Following are some of the ways to instantiate a java String
String str1 = "Hello";
String str2 = new String("Apple");
String str3 = new String(char []);

Methods:

There are methods for:

- concatenating two strings - concat() ,+

```
public class StringConcatWays {

    public static void main(String a[]) {
        String str1 = "TATA ";
        String str2 = "Consultancy Services";

        String str3 = "TATA ";
        String str4 = "Group of companies";
        /** We can do string concatination by two ways. * One is by using '+' operator, shown below.*/
        String d = str1 + str2;
        System.out.println(d);
        /** Another way is by using concat() method, * which appends the specified string at the end. */
        d = str3.concat(str4);
        System.out.println(d);

        /** You can also use the concat() method with string literals, as in: */
        String str5="My name is ".concat("Rumple");

        System.out.println(str5);
        /** The + operator is widely used in print statements.  */
        String string1 = "saw I was ";
        System.out.println("Dot " + string1 + "Tod");

    }
}
```

Result:

```
TATA Consultancy Services
TATA Group of companies
My name is Rumple
Dot saw I was Tod
```

- getting the character at a given position within the string -- charAt() .By using indexOf() method you get get the position of the sepcified string or char from the given string. You can also get the index strting from a specified position of the string.

ex:

charAt(int index)

@Returns the char value at the specified index

```
public class StringChatAt {
    public static void main(String args[]) {
        String s = new String("TATA Consultancy Services!");
        char result = s.charAt(6);
        System.out.println(result);
    }
}
```

will result as

```
o
```

Combining charAt() with another String method, we can get the final character of any String, regardless of length:

```
public class StringChatAt {
    public static void main(String args[]) {
        String s = new String("What a long, strange trip it's been.");
        char result = s.charAt(s.length()-1);

        System.out.println(result);


    }
}
```

will result as

- seeing if a character or string exists within a string -- indexOf()

By using indexOf() method you get get the position of the sepcified string or char from the given string. We can also get the index strting from a specified position of the string.

```
public class StringIndexOf {
    public static void main(String[] a) {
        String str = "Use this string for testing this";
        System.out.println("Basic indexOf() example");
        System.out.println("Char 's' at first occurance: " + str.indexOf('s'));
        System.out.println("String \"this\" at first occurance: "
                + str.indexOf("this"));
        /** * Returns the first occurance from specified start index */
        System.out
                .println("First occurance of char 's' from 4th index onwards : "
                        + str.indexOf('s', 4));
        System.out
                .println("First occurance of String \"this\" from 6th index onwards: "
                        + str.indexOf("this", 6));
    }
}
```

Output is

```
Basic indexOf() example
Char 's' at first occurance: 1
String "this" at first occurance: 4
First occurance of char 's' from 4th index onwards : 7
First occurance of String "this" from 6th index onwards: 28
```

- getting the number of characters in a string -- length()

Example:

Output:

```
Length:17
```

- extracting a substring from a string – substring()

The substring begins with the character at the specified index and extends to the end of this string or up to endIndex - 1 if second argument is given.

Syntax of this method:

public String substring(int beginIndex)

or

public String substring(int beginIndex, int endIndex)

Note:

beginIndex -- the begin index, inclusive.

endIndex -- the end index, exclusive.

Example:

```
public class SubstringExample {

    public static void main(String args[]) {
        String str = new String("TATA Consultancy Services");

        System.out.println("Initial string is: " + str);

        System.out.println("Start position=4 and no end position specified: "
                + str.substring(4));

        System.out.println("Start position=2 and end position=11: "
                + str.substring(2, 11));

        // if start = end, then the extracted string will be empty
        System.out.println("Start position=3 and end position=3: "
                + str.substring(3, 3).isEmpty());

    }

}
```

Output:

```
Initial string is: TATA Consultancy Services
Start position=4 and no end position specified:  Consultancy Services
Start position=2 and end position=11: TA Consul
Start position=3 and end position=3: true
```

Ask a doubt *(Misuse of 'Ask a Doubt' Section will be dealt as per the Terms & Conditions of Campus Commune)*

TATA CONSULTANCY SERVICES

Java Lounge

☐Logout

Java

1.IT Application Overview And Features	Q
2.Need of Programming and Introduction to OOP approach	Q
3.Introduction to Java,JVM,JDK	Q
4.Operators and Variables in Java	Q
5.Introduction to class in java,access specifiers, this and static in Java	Q
6.Java Library, Packages, Use of import	Q
7.Conditional operations	Q
8.Basic Iterations and Arrays	Q
9.Designing Web Pages Part 1(HTML and CSS)	Q
10.Designing Web Pages Part 2 (JavaScript and JQuery)	Q

☐ Course Completion Quiz

4.6. Reference Links & videos

Additional videos are listed below

Related Videos

Ask a doubt *(Misuse of 'Ask a Doubt' Section will be dealt as per the Terms & Conditions of Campus Commune)*

4.7. Practice Problems

1. Write a program to find the difference between sum of the squares and the square of the sums of 2 numbers?
2. Develop a program that accepts the area of a square and will calculate its perimeter.
3. Utopias tax accountants always use programs that compute income taxes even though the tax rate is a solid, never-changing 15%. Write a program that calculates net pay of an employee based on number of hours the employee has worked. Assume an hourly rate of $12.

4. An old-style movie theater has a simple profit program. Each customer pays $5 per ticket. Every performance costs the theater $20, plus $.50 per attendee. Develop the program that accepts the number of attendees (of a show) and calculates how much income the show earns.

5. Develop the program which computes the height that a rocket reaches in a given amount of time. If the rocket accelerates at a constant rate g, it reaches a speed of g • t in t time units and a height of 1/2 * v * t where v is the speed at t.

6. Develop a program that accepts the length and width of a rectangular floor and the edge length of a square tile and will compute the whole number of tiles needed to cover the floor completely.

Click here to download the Solutions

Ask a doubt *(Misuse of 'Ask a Doubt' Section will be dealt as per the Terms & Conditions of Campus Commune)*

Open Doubts | **Closed Doubts**

SASIVINEETHA GATTUPALLE

In the solution to the first problem the numbers taken are integers and the result would surely be an integer.Then why is the result stored in a variable of double data type instead of int data type? Thank You.

about 24 days ago

Java

Course Completion Quiz

5.Introduction to class in java,access specifiers, this and static in Java

Objective

- Introduction to Classes
- Declaration of Classes
- Declaration of instance variables
- Declaration of methods
- Object and Object creation
- Access Modifiers
- Constructors
- Use of this Keyword
- Examples for "this" keyword
- Static keyword

5.3 Constructors and Access Specifiers

5.4 Use of this Keyword

5.5 Static Keyword

5.6 Practice Problems

5.7 Reference Links & videos

5.1. Introduction to Classes

A class is a template, blueprint,or contract that defines what an object's data fields and methods will be. An object is an instance of a class. You can create many instances of a class.
A Java class uses variables to define data fields and methods to define actions. The attributes and behaviours defined by a class are common to all objects of the same kind.

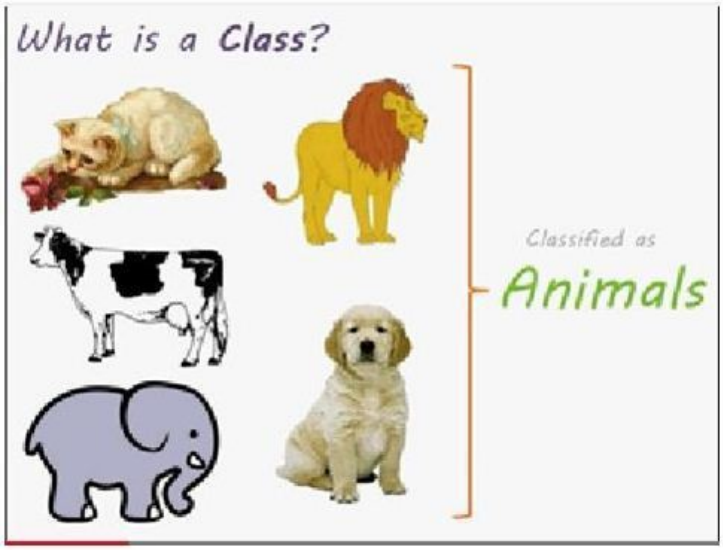
Declaration of Classes

A class is declared by the use of the "class" keyword. Class body is enclosed between curly braces { and }. The data or variables defined within the class are called instance variables. The code is contained within methods. Collectively, the methods and variables defined in a class are called members of the class.

Finally we can say a class is a blue print of particular classification of objects.

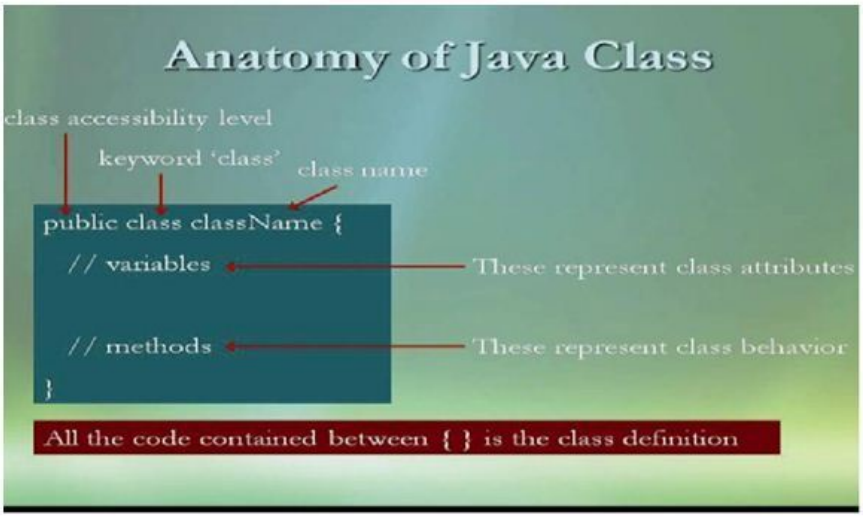
Please refer to the below image for more details.

Here we can say there is a class **Car**.



Here we can say there is a class called **Animal**..

- ✓ A Class is a blue print of a particular classification of Objects
- ✓ An Object belongs to a Class of Objects
- ✓ An Object is an instance of a Class
- ✓ For example:
  - ➢ myCar, petersCar are instances of a class called Car
  - ➢ myCat, petersDog are instances of a class called Animal

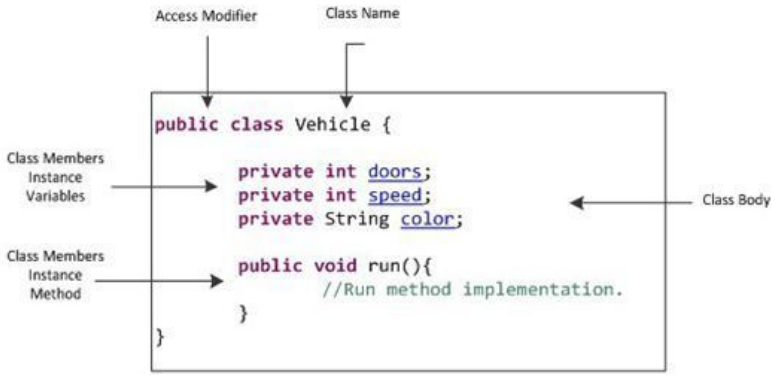


A class can contain any of the following variable types.

Local variables: Variables defined inside methods, constructors or blocks are called local variables. The variable will be declared and initialized within the method and the variable will be destroyed when the method has completed.

Instance variables: Instance variables are variables within a class but outside any method. These variables are instantiated when the class is loaded. Instance variables can be accessed from inside any method, constructor or blocks of that particular class.

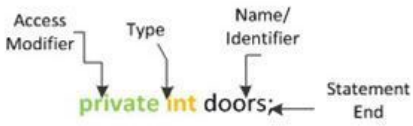
Class variables: Class variables are variables declared with in a class, outside any method, with the static keyword.



Declaration of Instance Variables

Variables defined within a class are called instance variables because each instance of the class (ie, each object of the class) contains its own copy of these variables. Thus, the data for one object is seperate and unique from the data of another. Instance variables can be declared public or private or default (no modifier). When we do not want our variables value to be changed outside our class we should declare them private. Public variables can be accessed and changed from outside the class. We will have more information about this on OOP tutorial.

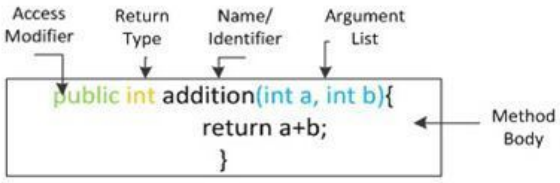
Below find the syntax for declaring an instance variable.

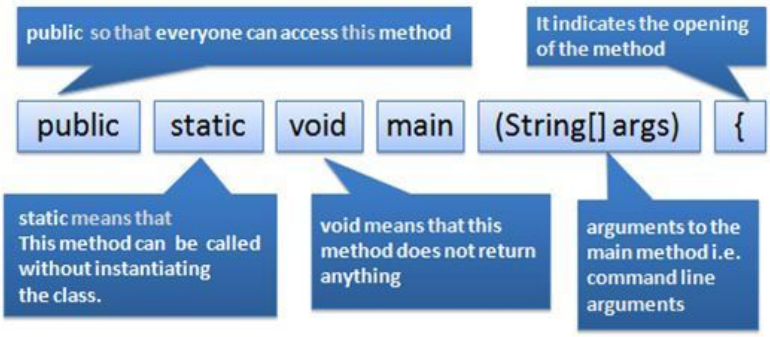


Declaration of Methods

A method is a program module that contains a series of statements that carry out a task. To execute a method, you invoke or call it from another method; the calling method makes a method call, which invokes the called method. Any class can contain an unlimited number of methods, and each method can be called an unlimited number of times.

The syntax for method declaration is as given below. Java main method is given as an example.





Ask a doubt *(Misuse of 'Ask a Doubt' Section will be dealt as per the Terms & Conditions of Campus Commune)*

Open Doubts | **Closed Doubts**

SHYAM VASANI

What is the last date for completion of Tech Lounge section?

about 17 days ago

SHABU K V **Moderator**

Day before you join TCS. In fact ILP will be a continuation of tech Lounge and better you prepare Tech Lounge easier it is to do ILP

SIDDHARTH GOSALIA

Should a java class name necessarily begin with an upper case letter? Is it a rule?

about 26 days ago

KALYAN PILLA

All PUBLIC class names begin with an upper case letter by "convention".

SAI KANUPARTHI

It is not a rule but we follow to understand conventions easily.

ABHISHEK SAH

It is not a rule,i.e you will not get an error if your class name does not start with upper case letter but Java has some naming "convention"/recommendations which is followed mostly for the easy of developer too.

You can get all the Java naming conventions: http://www.javacodegeeks.com/2011/08/java-naming-conventions.html

SHANKAR PAUL

Why string is given as arguments to the main function in java?

about 2 months ago

PREETHI PATTABIRAMAN

Well it is how the method signature is defined for main in jvm. You try as int args. ... it will compile but won't run. Because main doesn't match the signature as defined in jvm. String is the most common object in java.

PREETHI PATTABIRAMAN

https://www.google.co.in/url?sa=t&source=web&rct=j&ei=ZdaFU7uxJMSkkQXTuYGACQ&url=http://stackoverflow.com/questions/10183611/usin int-instead-of-string-public-static-void-main-int-args&cd=1&ved=0CCcQFjAA&usg=AFQjCNFXSz7LpKgUNoCiI2Dux7dcKO1rVA&sig2=kAyqyVw8w1AY7UwW1cyJYw

LAKSHMINADHA PRASAD THOTA

The main method of java takes String array i.e **String[] args** as parameter not **String.**

arguments are helpful to pass parameters to the main program

when you are trying to run a java program, JVM will search the main method with **String array** as argument to start the execution of the program from there. As the method you are given is not with that signature, so it will raise an exception **No main method found**

ABHISHEK SAH

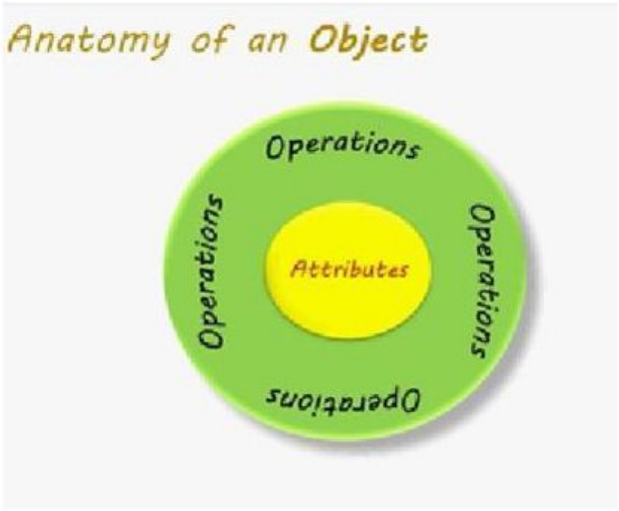
The String []args is an array of string which is passed to main. args is the array name here and basically it is helpful in taking "**command line arguments**". Command line arguments are values passed when you type the run command for the program. Eg: **c:/>** java Test 23 45

Here 23 and 45 are stored in this String array and can be used in the program.(Test here being the class name.)

5.2. Objects or Instances

Consider Tiger from the above list,

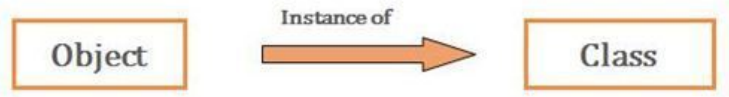
Objects are made up of attributes(properties) and methods(behaviours). Attributes are the characteristics that define an object; the values contained in attributes differentiate the objects of the same class from one another.

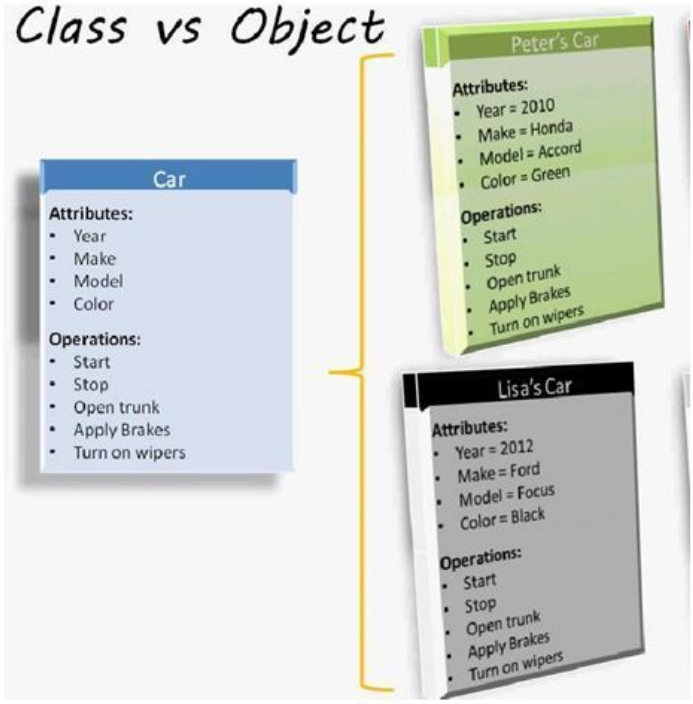


To understand this in a better way, let's take an example of **Mobile** as an object. Mobile has characteristics like model, manufacturer, cost, operating system etc. So if we create "Samsung" mobile object and "iPhone" mobile object, we can distinguish them from their characteristics. The values of the attributes of an object are also referred to as the **object's state**.

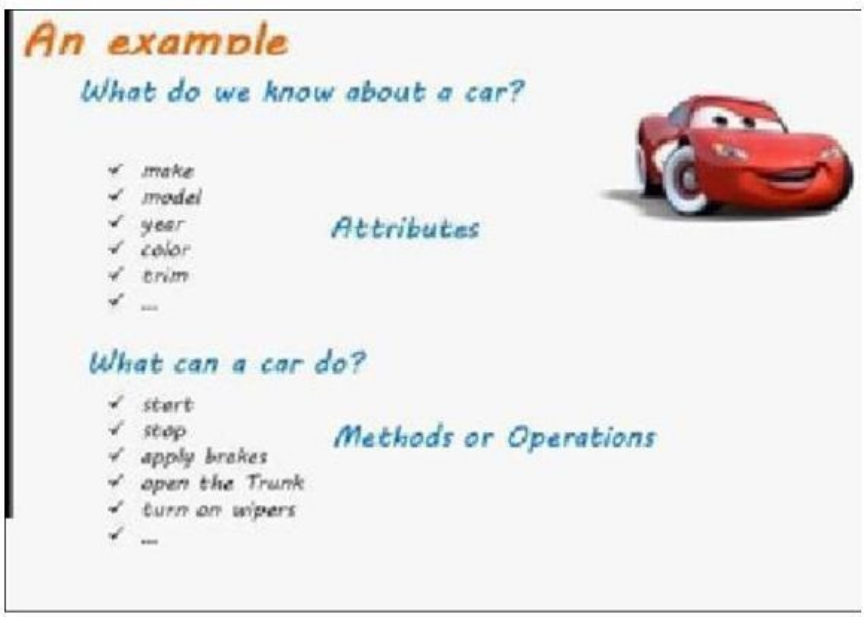
Objects need not be physical objects. It can be any entity. It can be account,an organization etc.

We can say --- *Object is an instance of a class*.



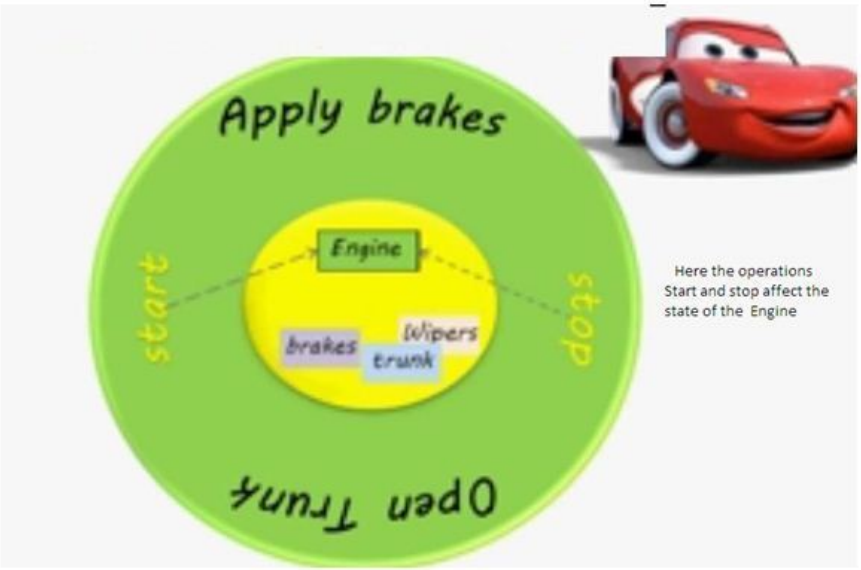


Consider an object Car. What we know about a car defines its attributes. What a car can do defines its behaviours/methods.



Always operations can alter the state of the attributes.

Consider the above start operation for a Car. When the car is started ,it is affecting the state of the engine.



Creating an Object:

As mentioned previously, a class provides the blueprints for objects. So basically an object is created from a class. In Java, the new key word is used to create new objects.

There are three steps when creating an object from a class:

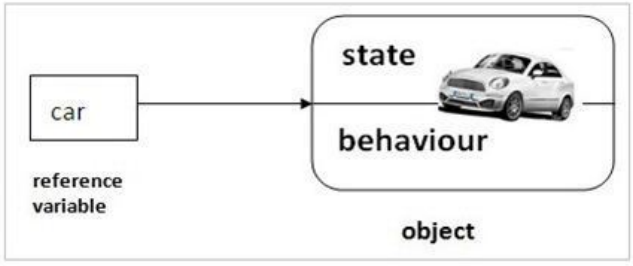
Declaration: A variable declaration with a variable name with an object type.

Instantiation: The 'new' key word is used to create the object.

Initialization: The 'new' keyword is followed by a call to a constructor. This call initializes the new object.

Example of creating an object is given below:

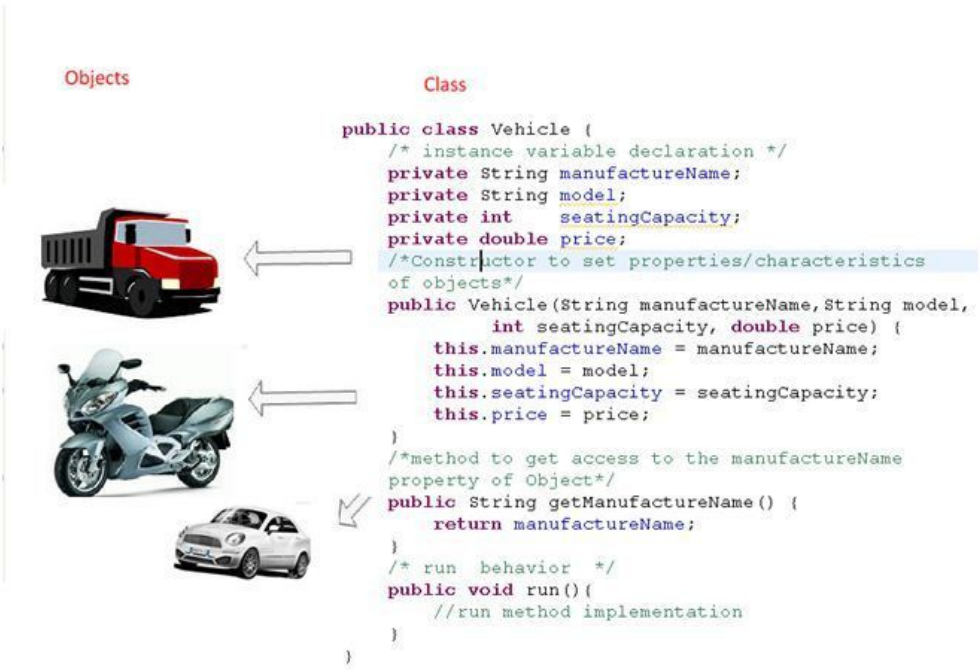
```
ClassName  ObjectReferenceVariableName = new ConstructorName();
Vehicle    car= new   Vehicle();
```



Refer the below Vehicle class.

We are creating three objects truck, bike and a car from Vehicle class.

```
//creating a truck object from Vehicle class
Vehicle truck= new Vehicle("Tata","T200-Big", 4,500000);

//creating  bike object from Vehicle class
Vehicle  bike=new Vehicle("Suzuki" "RX500" 2, 300000);

//Create car object
.....
.....
```



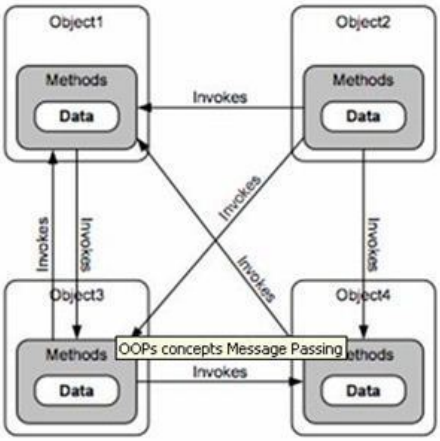
Accessing Instance Variables and Methods using Objects:

/* call a variable as follows */
ObjectReference.variableName;

/* Now you can call a class method as follows */
ObjectReference.MethodName();

How objects communicates each other?

One object invoking methods on another object is known as **Message passing**.
It is also referred to as **Method Invocation**.



Ask a doubt *(Misuse of 'Ask a Doubt' Section will be dealt as per the Terms & Conditions of Campus Commune)*

Open Doubts | **Closed Doubts**

SHRUTI SINHA

What is meant by 'make' attribute of car class?? What does it actually supposed to store?

about 22 days ago

SASIVINEETHA GATTUPALLE

What does the phrase "Methods of an object" exactly mean? Thank You.

about 24 days ago

LAKSHMI SHANKHWALKER

In a class you define variables and functions. Like class Car above, model is a variable and start() is a function. these functions are called as methods. when you create an object of the class say Alto, Alto has a model and a start() method. method is some function an object can do, or can have done to it.

SRIKAVYA PINGALI

Class is a keyword....I marked it as true but marks got deducted.....Please tell me if I m wrong...!!!

about 1 month ago

ASHUTOSH JHA

Its "class"..I guess you did not notice the caps C

SARANYA P

Can we create instance variables without access specifiers? If yes, what is the major difference between creating instance variables with and without access specifiers??

ie. Instance variable without public or private.

about 1 month ago

AMARDEEP BURNWAL

instance variable without access specifier is considered as 'default' access specified.

ANAND NARLA

Can we create a class without a **constructor?**

about 2 months ago

HARISH VALUROUTHU

yes anand we can create a class without a constructor

LAKSHMINADHA PRASAD THOTA

Yes in java we can create a class with out a constructor ...in this case **java compiler** will provide a default constructor(a constructor with out having any parameters) at the time of compilation.

ALURI REDDY

YES

If you don't specify a constructor, a default constructor will be generated by the compiler.

SHANKAR PAUL

what is the use of **this** keyword?

about 2 months ago

PREETHI PATTABIRAMAN

This keyword is used as a default object reference which refers the current object. This is used in constructor to initialize variables or in methods to indicate the data with which the method should be executed. Default object reference to current object is this.

PREETHI PATTABIRAMAN

This keyword is used as a default object reference which refers the current object. This is used in constructor to initialize variables or in methods to indicate the data with which the method should be executed. Default object reference to current object is this.

LAKSHMINADHA PRASAD THOTA

The keyword **this** is always represents the current object,that means inside class Test if i am using this keyword that represents the object of Test.

example:-

```
class Test{

int age;
String name;

 public Test(int age,String name){

  this.age=age;
  this.name=name;
 }

 public static void main(String[] args){

  Test t=new Test(28,"Ramu")

 }

}
```

in the above example age and name are instance varibles.
the constructor Test takes two parameters age and name same as instance varible.
To differentiate the instance variables with constructor parameters we have to specify the current object using **this** keyword. that means **this.age** refers the instance variable age but not constructor parameter age. and also **this.name** refers the instance variable name but not constructor parameter name.

hope this one is clear!

ALURI REDDY

In java, this is a reference variable that refers to the current object.

Usage of this keyword

1)this keyword can be used to refer current class instance variable.
2)this() can be used to invoke current class constructor.
3)this keyword can be used to invoke current class method (implicitly)

5.3. Constructors and Access Specifiers

Constructors

Java constructors are special methods which are used to initialize objects. Constructor method has the same name as that of class, they are called or invoked when an object of class is created and can't be called explicitly. They are designed to perform initializing actions such as initializing the data fields or objects.

A java constructor has the same name as the name of the class to which it belongs. Constructor's syntax does not include a return type, since constructors never return a value.

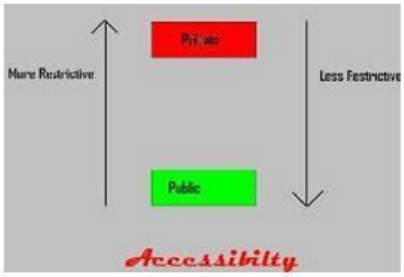
Constructors can be classified into two types, default constructors and parametrized constructors.
If you don't define a constructor, then the compiler creates a default constructor. Default constructors do not contain any parameters. Default constructors are created only if there are no constructors defined by us. Java provides a default constructor which takes no arguments, when no explicit constructors are provided. Parametrized constructors are required to pass parameters on creation of objects.

```
class Vehicle {
    /* instance variable declaration */

    private String manufactureName;

    private String model;

    /*default constructor */
    public Vehicle(){
        System.out.println("Default Constructor called");
    }
    /* Constructor to set properties/characteristics of objects */
    public Vehicle(String manufactureName, String model){
        System.out.println("Constructor with arguments invoked");
        this.manufactureName=manufactureName;
        this.model=model;
    }

    /* main method.Execution starts from main method */
    public static void main(String[] args) {
        // creating   vehicle object.Calling constructor to initialize
        // the instance variables.
        Vehicle  car= new Vehicle();
        Vehicle  bus= new Vehicle("Tata", "Air bus-T200");

    }

}
```

```
Console
Default Constructor called
Constructor with arguments invoked
```

Access Specifiers

Each object has members (members can be variables or methods) which can be declared to have specific access.



Access Modifier - private:

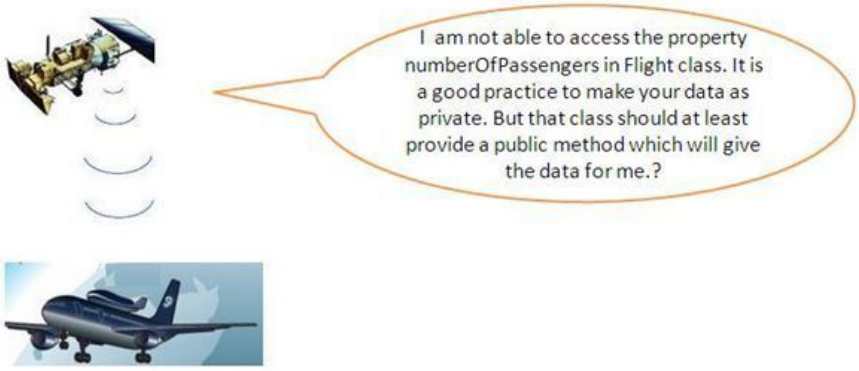
Methods, Variables and Constructors that are declared private can only be accessed within the declared class itself. Private access modifier is the most restrictive access level.

Variables that are declared private can be accessed outside the class if public getter methods are present in the class. Using the private modifier is the main way that an object encapsulates itself and hide data from the outside world.

Access Modifier - public:

Members (variables, methods and constructors) declared public (least restrictive) within a public class are visible to any class in the java program, irrespective of whether these classes are in the package or in another package.

```
public class Flight {

    private String flightNumber;
    private String origin;
    private String destination;
    private int numberOfPassengers;

    public String getFlightNumber() {
        return flightNumber;
    }

    // other behaviors
}/* Flight */
```



Below program will give compilation Error.

```
public class Satellite {

    public static void main(String[] args) {
        /* Satellite class trying to access the numberOfPassengers in flight */
        Flight flight1= new Flight();
        int numberOfPassengers=flight1.numberOfPassengers;/* this property not accessibl
        System.out.println("Number of passengers in flight1 is:"+numberOfPassengers);
    }

}
```

```
public class Flight {

    private String flightNumber;
    private String origin;
    private String destination;
    private int numberOfPassengers;

    public String getFlightNumber() {
        return flightNumber;
    }
    public int getNumberOfPassengers() {
        return numberOfPassengers;
    }

    // other behaviors
}/* Flight */
```

```
public class Satellite {

    public static void main(String[] args) {
        /* Satellite class trying to access the numberOfPassengers in flight */
        Flight flight1= new Flight();
        /*Now using getNumberOfPassengers()  method Satellite  can access
          numberOfPassengers in flight.Always make your data as
          private and provide getter and setter methods  to access the data */
        int numberOfPassengers=flight1.getNumberOfPassengers();
        System.out.println("Number of passengers in flight1 is:"+numberOfPassengers);
    }

}
```

Ask a doubt *(Misuse of 'Ask a Doubt' Section will be dealt as per the Terms & Conditions of Campus Commune)*

Open Doubts | **Closed Doubts**

ABHISHEK SAH

Which access specifier is most suitable for constant values? (Asked in the quiz.)

about 17 days ago

MADDI

static is the most suitable one because its value is shared by all the objects.

SIDDHARTH GOSALIA

"Variables that are declared private can be accessed outside the class if public getter methods are present in the class." What does this statement mean?

about 26 days ago

VAIBHAV CHAUDHARY

As u can see in above example Satellite class can access "number of passengers" variable which is declared private in Flight class with the help of "getNumberOfPassengers" method in Flight class which is declared public in Flight class. As per the rule variables declared with private access specifiers can be accessed only with in the class so if some other class wants to access a private variable they must access that variable using a public method of that class (i.e in case of above problem using getNumberOfPassengers method).That means you can not access the private variable directly but you can access it with the help of public method in the same class.

MOHAMMED RAHIMAN

questn:: ***Which of these is used as default specifier for a member of class if no access specifier is used for it ???***

a) private
b) public
c) public, within its own class
d) protected

I answered (a)....since default access specifier for class members is private(strictly speaking package-private)...though it deduced my marks..

similarly,, questn::**Which of these is used to access member of class before object of that class is created**?
a) public
b) private
c) static
d) protected

here..i went with (c),,since for static members can be accesed without any object creation and instantiation,,where we can call the static members prefixed with its classname.....finally it too turned out be the wrng answer.....

are they really wrong answers..plzzz let me kno abt your ideas on those questns.....

THANXin advance...

about 2 months ago

ANBU SELVAM

I also suffered with the same problem...

LAKSHMINADHA PRASAD THOTA

For the first questions there is no suitable answer in that list...as you mentioned **private is not a package-private**, **default(no modifier) is the package-private.**

For the second question **static** is the correct answer.

PRIYANKA SAWANT

the ans to the first q is **public ,within its own class.**

bcz default access specifier allows a class to be accessed by other classes within the same package.

DIVYA PANICKER

Answer for the first question is "public within its own **package**". When default specifier is used for the member of a class it can be accessed from any class within the same package.

Answer for the second question is "static"

SHANKAR PAUL

As mentioned earlier there are four access specifiers in java.What is the fourth access specifier and what are its properties?

about 2 months ago

PREETHI PATTABIRAMAN

The fourth is simply referred as default. Public private protected and default. But while I was attending a class my teacher called the default specifier as "friendly"

PREETHI PATTABIRAMAN

http://javapapers.com/core-java/access-modifiers-in-java-explain/

LAKSHMINADHA PRASAD THOTA

In java there are four Access Modifiers are available

1)public String name;

2)private String name;

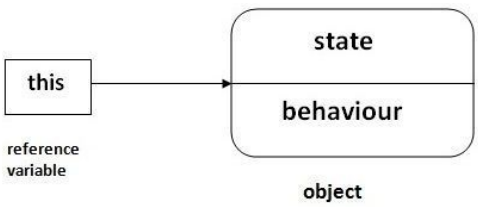
3)protected String name;

4)String name; // this is the **default** one we can not specify any thing.

the default can access in the same class and other classes which are in same package. it is also called **'package-private' or 'default' or 'friendly' access? modifier.**

5.4. Use of this Keyword

There can be a lot of usage of **this** keyword. In java, **this** is a reference variable that refers to the current object.



Example on this keyword

If there is ambiguity between the instance variable and parameter, this keyword resolves the problem of ambiguity.

Understanding the problem without this keyword

Let's understand the problem if we don't use this keyword by the example given below:

```
class Vehicle {
    /* instance variable declaration */
    private String manufactureName;
    private String model;
    private int    seatingCapacity;
    private double price;
    //Constructor to set properties/characteristics of objects
    public Vehicle(String manufactureName,String model,
            int seatingCapacity, double price) {
        manufactureName = manufactureName;
        model = model;
        seatingCapacity = seatingCapacity;
        price = price;
    }
    /*  main method.Execution starts from main method */
    public static void main(String[] args) {
        //creating a vehicle object.Calling constructor to initialize
        //the instance variables.
        Vehicle  car= new Vehicle("Tata", "Indica Vista", 5, 450000);
        System.out.println("manufactureName ---:"+car.manufactureName);
        System.out.println("model ---:"+car.model);
        System.out.println("seatingCapacity ---:"+car.seatingCapacity);
        System.out.println("price ---:"+car.price);
    }
}
```

```
Console
manufactureName ---:null
model ---:null
seatingCapacity ---:0
price ---:0.0
```

In the above example, parameter (formal arguments) and instance variables are same that is why we are using this keyword to distinguish between local variable and instance variable.

Solution of the above problem by this keyword

```
class Vehicle {
    /* instance variable declaration */
    private String manufactureName;
    private String model;
    private int    seatingCapacity;
    private double price;
    /*Constructor to set properties/characteristics of objects */
    public Vehicle(String manufactureName,String model,
            int seatingCapacity, double price) {
        this.manufactureName = manufactureName;
        this.model = model;
        this.seatingCapacity = seatingCapacity;
        this.price = price;
    }
    /*  main method.Execution starts from main method */
    public static void main(String[] args) {
        //creating a vehicle object.Calling constructor to initialize
        //the instance variables.
        Vehicle  car= new Vehicle("Tata", "Indica Vista", 5, 450000);
        System.out.println("manufactureName ---:"+car.manufactureName);
        System.out.println("model ---:"+car.model);
        System.out.println("seatingCapacity ---:"+car.seatingCapacity);
        System.out.println("price ---:"+car.price);
    }
}
```

this keyword used

```
Console
manufactureName ---:Tata
model ---:Indica Vista
seatingCapacity ---:5
price ---:450000.0
```

Solution of the above problem by with out using this keyword

```
class Vehicle {
    /* instance variable declaration */
    private String manufactureName;
    private String model;
    private int    seatingCapacity;
    private double price;
    /*Constructor to set properties/characteristics of objects */
    public Vehicle(String manufactureNameValue,String modelValue,
                   int seatingCapacityValue, double priceValue) {
        manufactureName = manufactureNameValue;
        model = modelValue;
        seatingCapacity = seatingCapacityValue;
        price = priceValue;
    }
    /*  main method.Execution starts from main method */
    public static void main(String[] args) {
        //creating a vehicle object.Calling constructor to initialize
        //the instance variables.
        Vehicle  car= new Vehicle("Tata", "Indica Vista", 5, 450000);
        System.out.println("manufactureName ---:"+car.manufactureName);
        System.out.println("model ---:"+car.model);
        System.out.println("seatingCapacity ---:"+car.seatingCapacity);
        System.out.println("price ---:"+car.price);
    }
}
```

Console

```
manufactureName ---:Tata
model ---:Indica Vista
seatingCapacity ---:5
price ---:450000.0
```

Ask a doubt *(Misuse of 'Ask a Doubt' Section will be dealt as per the Terms & Conditions of Campus Commune)*

Open Doubts | **Closed Doubts**

AMALA VANKAYALA

How the problem got solved without using this keyword. what is the function of the value

about 2 months ago

PREETHI PATTABIRAMAN

It's just a variable name. You can even name it x y z etc. And it'll work fine.

TEJASWINI

This is because the instance variables used in the class Vehicle are not same as that of the parameter names in the constructor Vehicle.

5.5. Static Keyword

Static variables are also known as Class variables. Static variables are declared with the static keyword in a class, but outside a method, constructor or a block.

There would only be one copy of each class variable per class, regardless of how many objects are created from it.Static variables are stored in static memory.

Static variables are created when the program starts and destroyed when the program stops.

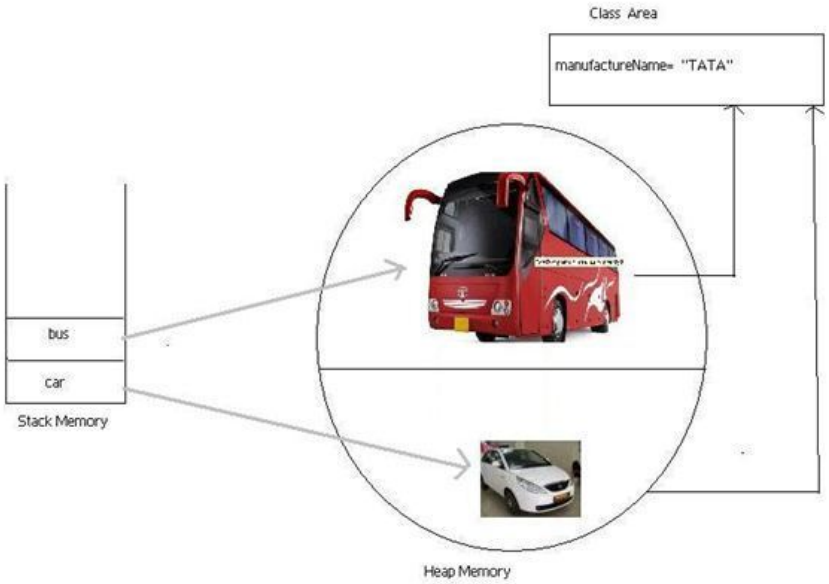
Visibility is similar to instance variables. However, most static variables are declared public since they must be available for users of the class.

Default values are same as instance variables. For numbers, the default value is 0; for Booleans, it is false; and for object references, it is null. Values can be assigned during the declaration or within the constructor.

Static variables can be accessed by calling with the class name . ClassName.VariableName.

Consider the below class Vehicle, here manufacturteName is declared as static and getManufactureName() method also declared as static.

```
class Vehicle {
    private static String manufactureName="TATA"; /* class variable */
    private String model;
    private String color;
    private int    seatingCapacity;
    private double price;
    public Vehicle(String model,String color,
            int seatingCapacity, double price) {
        this.model = model;
        this.color=color;
        this.seatingCapacity = seatingCapacity;
        this.price = price;
    }
    /*method getManufactureName() is declared as static.Class Behavior*/
    public static String  getManufactureName(){
        return manufactureName;
    }
    public static void main(String[] args) {
        Vehicle  car= new Vehicle("Indica Vista","White", 5, 450000);
        System.out.println("manufactureName ---:"+Vehicle.getManufactureName());
        System.out.println("model ---:"+car.model);
        Vehicle  bus= new Vehicle("Tata Divo","Red", 30, 600000);
        System.out.println("manufactureName ---:"+Vehicle.getManufactureName());
        System.out.println("model ---:"+bus.model);
    }
}
```



OUTPUT is:

```
Console
manufactureName ---:TATA
model ---:Indica Vista
manufactureName ---:TATA
model ---:Tata Divo
```

Ask a doubt *(Misuse of 'Ask a Doubt' Section will be dealt as per the Terms & Conditions of Campus Commune)*

Open Doubts | **Closed Doubts**

SASIVINEETHA GATTUPALLE

The class Vehicle is closed only after main method then the private variables are available to the main method also. Now what is the need of the method "getManufactureName" in the program? Thank you.

about 24 days ago

KALYAN PILLA

The method getManufactureName() is not of much importance in the given example. However, if you were to access the variable 'manufactureName' from another class (eg., if your main was in another class) the method getManufactureName() would be necessary.

BSANJAYA

anyone completed programs and uploaded??

about 1 month ago

Java Lounge

☐Logout

Java

1.IT Application Overview And Features	Q
2.Need of Programming and Introduction to OOP approach	Q
3.Introduction to Java,JVM,JDK	Q
4.Operators and Variables in Java	Q
5.Introduction to class in java,access specifiers, this and static in Java	Q
6.Java Library, Packages, Use of import	Q
7.Conditional operations	Q
8.Basic Iterations and Arrays	Q
9.Designing Web Pages Part 1(HTML and CSS)	Q
10.Designing Web Pages Part 2 (JavaScript and JQuery)	Q

☐ Course Completion Quiz

5.6. Practice Problems

Note:

All attributes to be private for below classes.

Declare public getters and setters as per required.

Declare constructor which takes required parameters.

1. Declare class Calculator with attributes:

double valueOne, double valueTwo

Write below methods in this class and test those methods:

Add() //addes two values

Subtract() //subtracts valueTwo from valueOne

Multiply() // multiplies valueTwo and valueOne

Divide() //Divides valueTwo with valueOne. Returns 0 of valueOne is 0.

2. Declare class Triangle with below attributes:

double base, double height, double sideOne, doubleSideTwo

Write below methods in this class amd test those methods:

GetArea() // return area of the triangle
GetPeremeter() //return peremeter of the triangle

3. Declare class Cuboid with below attributes:

length, width, height

write below methods in this class and test those methods:

GetSurfaceArea() // return surface area of the cuboid
GetVolume() // return volume of the cuboid

4. Create class BankAccount with below attributes:

customerId, accountId,balance

write below methods in this class and test those methods:
DepositAmonut(double amount) // adds the amount to existing balance and returns new balance
WithdrawAmount(double amount) // deducts the amount from existing balance and returns new balance.
// however, if amount is greater than balance, returns -1
AddInterest(double percent) // adds amount based on interest percentage and returns new balance

5. Declare class Associate with below attributes:

id, grade (can be A/B/C), designation (can be J/M/S), basicPay
//basic pay for designation J = 10000, M = 15000, S = 20000

Write below methods:

PromoteEmployee(char newDesignation) //Check if new designation is not equal to or
less than current. If yes, return -1. else return 1 after promoted.
CalculateSalary()
Guidelines for CalculateSalary():

Grade A: HRA as 15%, DA as 25%
Grade B: HRA as 10%, DA as 20%
Grade C: HRA as 5%, DA as 15%

Designation: S : Add designation pay as 50% of basicPay
Designation: M : Add designation pay as 40% of basicPay
Designation: J : Add designation pay as 20% of basicPay

6. Declare class Time with below attributes:

hour, minute, second

Write below methods:

GetCurrentTime() //return Stirng as hh:mm:ss
Addtime(int hh, int mm, int ss) // adds seconds, minutes and hours to existing time and returns new time in hh:mm:ss format
SubtractTime(int hh, int mm, int ss) // subtracts seconds, minutes and hours to existing time and returns new time in hh:mm:ss format

Click here to download the solutions

Ask a doubt *(Misuse of 'Ask a Doubt' Section will be dealt as per the Terms & Conditions of Campus Commune)*

Java

Course Completion Quiz

6.Java Library, Packages, Use of import

Objective

a) Introduction to Packages and Imports

b) Understanding Java Libraries

c) More on access specifiers

6.4 Reference Links & videos

6.1. Packages

A package is a namespace that organizes a set of related classes and interfaces, providing access protection and name space management. Conceptually packages can be thought of as similar to different folders on the computer. Packages are used in Java to prevent naming conflicts, to control access, and to make searching/locating and usage of classes easier.

Java platform provides a very large library of classes and interfaces organized into different packages known as "Application Programming Interface" or API. Some of the packages in Java API Library are:

java.lang - bundles the fundamental classes

java.io - classes for input, output functions are bundled in this package

Because software written in Java can be composed of hundreds of individual classes, programmers can define their own packages to group related classes together. It is a good practice to organize by grouping related classes and interfaces into packages so that a programmer can easily locate and find the classes he needs.

Since the package creates a new namespace there won't be any class name conflicts with other class names in other packages.

Eg: Both java.awt and java.util packages have a class called "List". There is no naming conflict because the two "List" classes are part of two different packages.

Using packages, it is easier to provide access control and it is also easier to locate the related classes.

Eg: Consider a scenario where a bank has many employees. New employees are added, some may resign from bank, so those employees need to be removed from bank. Many customers come to the bank to get services. Customers open account, deposit money to account, withdraw money from account etc.

Here we have different objects and so different classes need to be created. If it is going to be a large application, it will be difficult to manage all the files without using packages. Also if we want to set different visibility, like Customer should not see the details of Employee, packages can be used to organize and set visibility using access modifiers.

Implementation of Packages in Java

The package in Java is represented using the keyword 'package'. The package statement is written with the keyword 'package' followed by the name of package and ends with a semi colon.

Syntax: package packagename ;

Eg: package employee ;

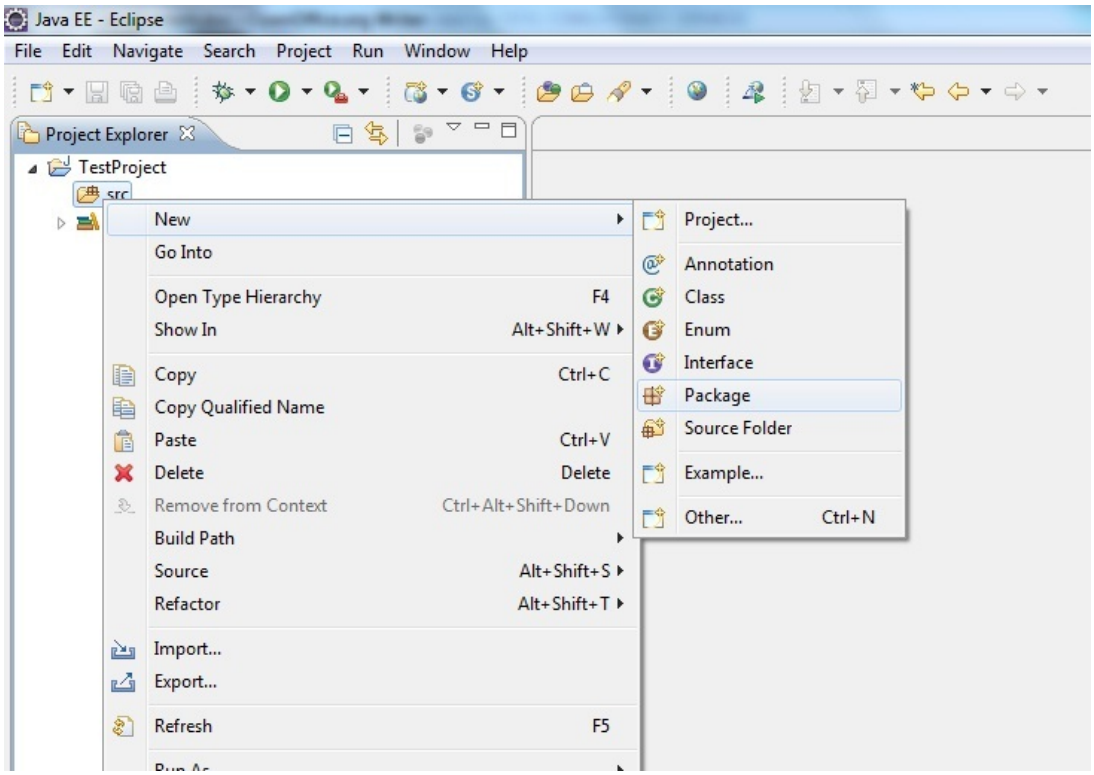
package com.tcs.employees ;

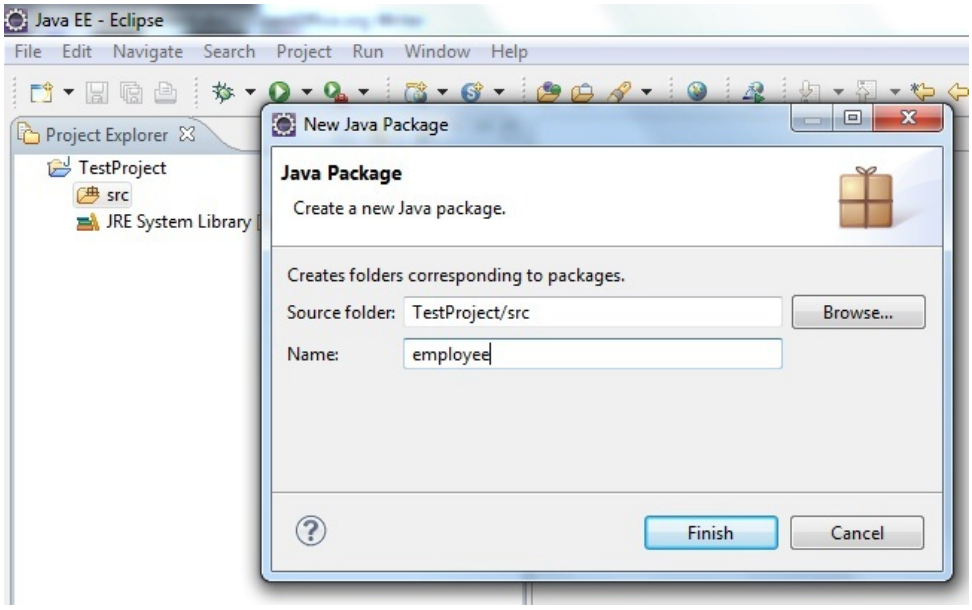
The package statement should be the first line in the source file. There can be only one package statement in each source file, and it applies to all types in the file.

Creating packages in Eclipse IDE

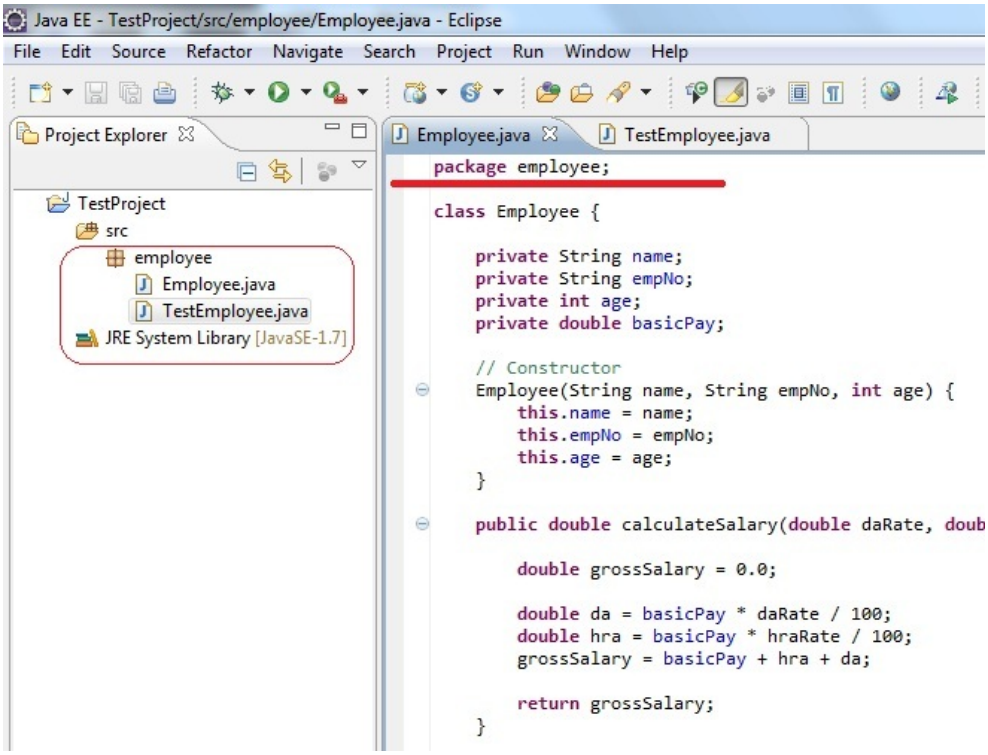
Step1: Create a package in eclipse and provide a name for the package as shown below. Package name can be a one word name or multiple words separated by a dot.

Eg: com.tcs.employees (or) employee





Step 2: Create classe(s) inside the employee package. The java class file should have package statement as the first line. In the screenshot below you can see the employee package. The classes Employee.java and TestEmployee.java are created under this package. You can see that the first line in Employee.java is the package statement.



Ask a doubt *(Misuse of 'Ask a Doubt' Section will be dealt as per the Terms & Conditions of Campus Commune)*

Open Doubts | **Closed Doubts**

DIVJYOT MAKKAR

Are static members visible outside the package?

What is static import?

What are wrapper classes?Is java.lang.void a primitive wrapper class?

about 16 days ago

SWARNADEEP SAHA

Hi,
Visibility and access does not depend on static keyword. To use the static member outside of the package it must be public.
like. public static <whatever>

SWARNADEEP SAHA

The normal import declaration imports classes from packages, so that they can be used without package reference. Similarly the static import declaration imports static members from classes and allowing them to be used without class reference. (source : http://javapapers.com/core-java/what-is-a-static-import-in-java/)

A wrapper class wraps primitive data type and give them an object appearence.

I think there are only 8 of that type. Byte, Short, Integer, Long, Float, Double. Charecter and Boolean. If you find that void is also one of them please let me know.

6.2. Import Statement

A package is to group related classes or other type of files together. The classes and other types under a package are called as package members. So as we saw above, we have grouped all files into different packages.

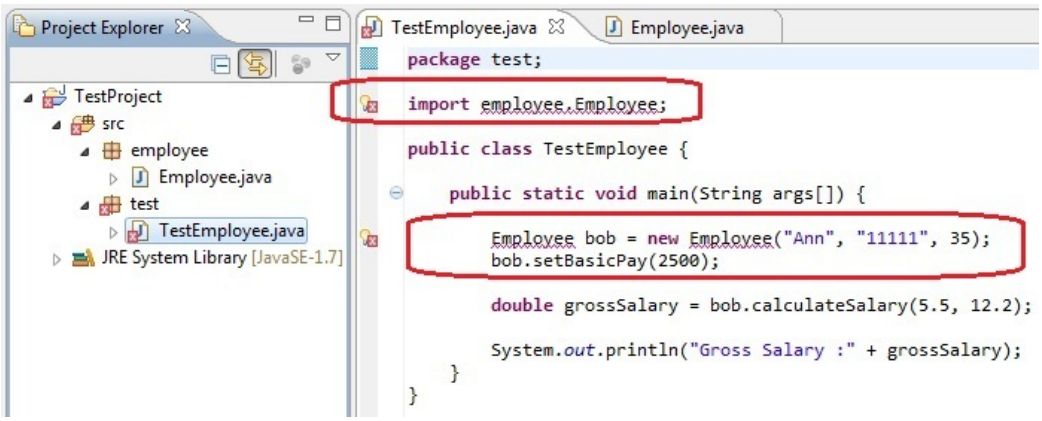
If a class wants to access/use another class/member from the same package, the class need not be imported. Classes/members within the same package can access each other without importing them.

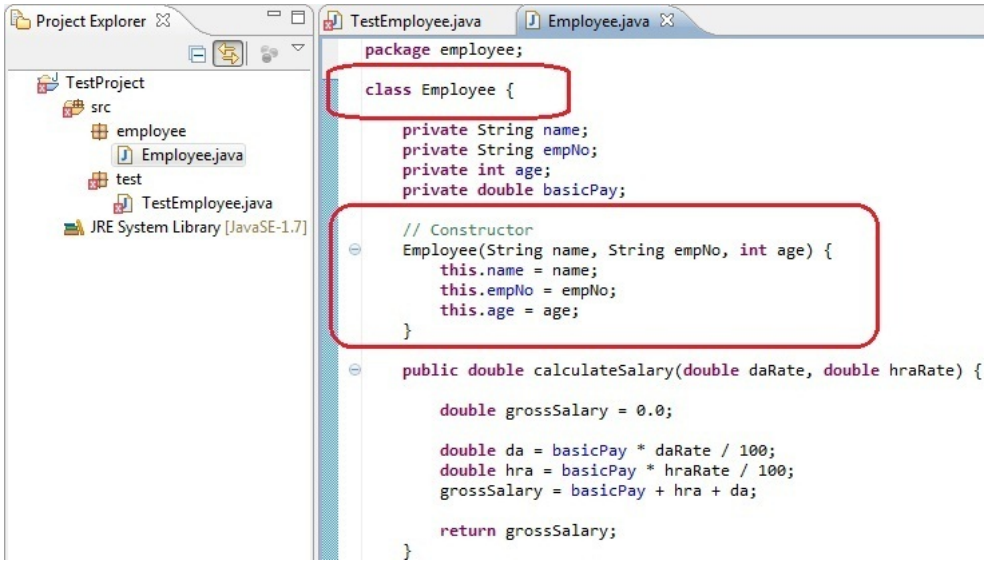
But in the scenario where the classes belonging to different packages need to access each other, the class belonging to the other package must be imported into the class that wants to use it.

The classes are imported using import statements. Import keyword is used to import the classes from different packages. Import keyword is followed by the fully qualified name of the class which includes the package name under which the class is present.

Eg: import java.util.Date ;

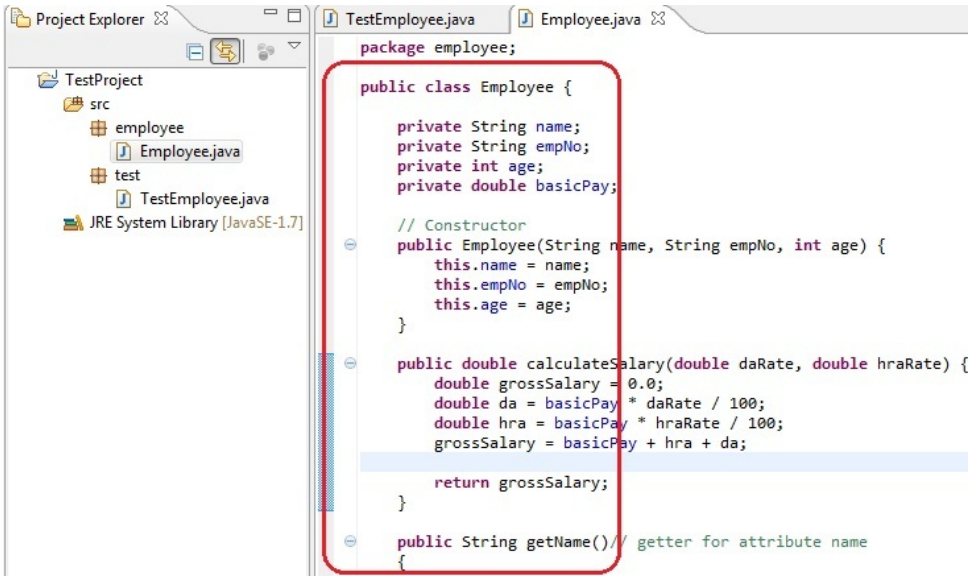
In the example below the Employee class is imported into the TestEmployee class.





You can see in the above example, the TestEmployee class is under the test package. You can also see the import statement to import Employee in TestEmployee class. This is there because TestEmployee is using Employee class, which is part of a different package employee. However even after the Employee class is imported into TestEmployee there are still errors in Testemployee class. This is because the Employee class is not visible to classes outside its package. Earlier when the TestEmployee class was inside employee package, it knew the Employee class. But when it was moved into a different package, and even after Employee class was imported the Employee class and its constructor are not visible. This is because only public members of a package/class are visible to classes from other packages. So specify public access specifier to the class, constructor and method which need to be accessed from outside.

In the below screen, you can see the changes made to access specifiers of Employee class and its constructor and methods. They are all declared with public access modifier.



To use a public package member from outside its package any one of the following can be used

a) Refer to the class/member by its fully qualified name

b) Import specific class/member in a package

c) Import all members/classes in a package

Refer to the class/member by its fully qualified name
employee.Employee = new employee.Employee() ;

Import all members/classes in a package:
By using * wild card, we can import all the classes inside employee package.

```
import employee.* ;
```

Import specific members/classes in a package:
We can also specify only the needed classes in the import statement.

```
import employee.Employee ;
```

If a class in a package has same name as another class in another package and both packages are imported then you must refer to each class by its qualified name to resolve any name ambiguities.

Ask a doubt *(Misuse of 'Ask a Doubt' Section will be dealt as per the Terms & Conditions of Campus Commune)*

Open Doubts | **Closed Doubts**

SURAJ V

Is it possible to import static classes from a package to another package?

about 11 days ago

SOURAV CHAKRABORTY

There isn't any setBasicpay method in Employee class. Is bob.setBasicpay(2500) a correct statement?

about 15 days ago

HALIMA SHAHUL

What is meant by fully qualified name.

about 2 months ago

ABUL M

If you import identically-named classes from different packages, you must use the fully qualified names when declaring the classes. For example, the Java core libraries contain the classes **java.sql.Date** and **java.util.Date**. In this case, you must write the fully qualified names of **java.sql.Date** and **java.util.Date** to use them.

Example : java.lang.* is used any class. but if we have two identical class name in two different package then we should use java.lang.Classname

LAKSHMINADHA PRASAD THOTA

Fully qualified name means including its package name where that class or interface is available

example:- For **String** class fully qualified name is **java.lang.String;**

For **OracleDriver** class fully qualified name is **oracle.jdbc.driver.OracleDriver.**

6.3. More on Access Modifiers and Java Libraries

We have discussed about 2 access specifiers in the previous courses. Which are they?

Yes -> public and private.

In Java, there are 4 access specifiers, they are:

Private Access Modifier – private

Methods, Variables and Constructors that are declared private can only be accessed within the declared class itself. Private access modifier is the most restrictive access level.

Variables that are declared private can be accessed outside the class if public getter methods are present in the class. Using the private modifier is the main way that an object encapsulates itself and hide data from the outside world.

Public Access Modifier – public

A class, method or constructor declared public can be accessed from any other class. Therefore attributes or methods declared inside a public class can be accessed from any other Java class which is inside the same package or inside another package. If the public class which we try to access is in

a different package, it needs to be imported first.

Default Access Modifier - No keyword

For default access modifier, we do not explicitly declare an access modifier for a class, field, method, constructors etc.

A variable or method declared without any access control modifier is available to any other class in the same package. Such members can be considered as package private.
We have already seen such a scenario when we discussed about import statements.

Eg: The constructor method in the following Java class is in default access.

```
class Employee {
    private String name;
    private String empNo;
    private int age;
    private double basicPay;
    // Constructor
    Employee(String name, String empNo, int age) {
        this.name = name;
        this.empNo = empNo;
        this.age = age;
    }
}
```

Protected Access Modifier – protected

Protected is related to inheritance (a parent child relation). Protected members in a class can be accessed only by its child classes from same/different package as well as by any other classes in the same package.

The child class can be anywhere, in other package or within the same package of the parent class.

The protected access modifier can be applied only to methods or fields, not to class. We can see more on it when we go through inheritance.

Java Libraries

Java differs from most other languages in that the number of classes and interfaces in its standard libraries is very large. Many common tasks have already been implemented by these libraries.

They are generally of high quality and are widely used. Implementing something which already exists in the libraries is probably wasted effort. Significant changes and additions to the standard libraries occur in each major release of Java, and it pays to keep current.

The most widely used packages are java.lang and java.util. There are other packages used for working with data such as java.sql, javax.sql, java.io etc.
For graphical applications, see the Swing classes (javax.swing, and so on).

You should go through JDK documentation just to get an idea of what is available. Later, when a specific need arises, you will often know which packages might be helpful.

Ask a doubt *(Misuse of 'Ask a Doubt' Section will be dealt as per the Terms & Conditions of Campus Commune)*

Open Doubts | **Closed Doubts**

SOLMANRAJU MAGANTI

What are Public Getter Methods ?

about 1 month ago

TEJASWINI

Public Getter Methods are the methods that take 0 parameters and returns a value. The returned value can be private member present in the class.

i.e we can access the private members present in a class through another class.

7.Conditional operations

Objective

- Understand conditional constructs, boolean operations and expressions, logical operators and precedence
- Perform simple, nested and complex conditional operations
- Ability to choose right conditional constructs

7.4 Reference Links & videos

TATA CONSULTANCY SERVICES

Java Lounge

☐Logout

Java

1.IT Application Overview And Features	Q
2.Need of Programming and Introduction to OOP approach	Q
3.Introduction to Java,JVM,JDK	Q
4.Operators and Variables in Java	Q
5.Introduction to class in java,access specifiers, this and static in Java	Q
6.Java Library, Packages, Use of import	Q
7.Conditional operations	Q
8.Basic Iterations and Arrays	Q
9.Designing Web Pages Part 1(HTML and CSS)	Q
10.Designing Web Pages Part 2 (JavaScript and JQuery)	Q

☐ Course Completion Quiz

7.1. Conditional Operations

There are many scenarios in our everyday life, where we need to take some decisions based on some criteria or condition. Decision making is one of the major part of Java also.

Conditional logic is involved when different operations are to be performed based on whether a decision is true or not. Conditional expressions consists of conditions that results in a boolean value and performs actions based on the boolean result. There are three types of decision making statements in Java. They are:

1. if – This statement is the most basic of all control flow statements. "if" statement has a certain condition expression. It executes a certain section of code only if that particular condition is true. If the condition is false, control jumps to the end of the if statement.

```
if(condition)
{
  /* statement(s) will execute if the condition is true.
    For example, a=5
```

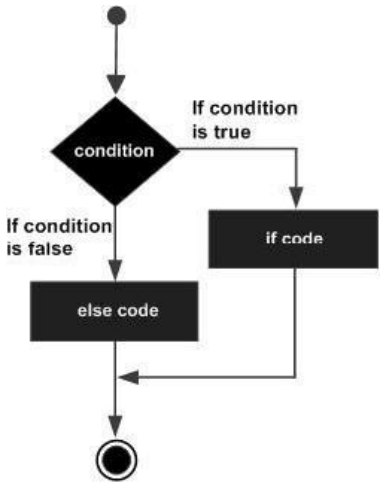
```
    Checking condition -
      if(a>3)
         {
          This code will be executed because 'a' is greater than 5.
    }
*/
}
```

2. If ... else – This statement provides a secondary path of execution when the condition becomes false. There can be nested if then else statement also.

```
if(condition)
   {
        /* statement(s) will execute if the condition is true.*/
   }
else
   {
       /* statement(s) will execute if the boolean condition is false */
   }
```

In the above example, if a=5 and condition is to check if a>6 then it will go to else block.

Flow Diagram:



3. switch – A switch statement allows a variable to be tested for equality against a list of values. Each value is called a case, and the variable being given as input is checked for each case.
This statement helps to select one out of the many choices based on an integer or String value. The conditions will accept only integer, character or String. The syntax is different from if else statements. Here we are comparing the input value with different cases. We can set a default case for inputs which does not satisfy any case.
The syntax of the switch statement is as follows.

```
    switch (expression) {
      case value_1 :
                  statement(s);
                  break;
      case value_2 :
                  statement(s);
                  break;
```

```
    .
    .
    .
    case value_n :
            statement(s);
            break;
    default:
            statement(s);
  }
```

Failure to add a break statement after a case will not generate a compile error but may have more serious consequences because the statements on the next case will be executed.

The following rules apply to a switch statement:

1. The variable used in a switch statement can only be a byte, short, int, char or String(It is allowed from jdk 1.7 version.).

2. You can have any number of case statements within a switch. Each case is followed by the value to be compared to and a colon.

3. The value for a case must be the same data type as the variable in the switch and it must be a constant or a literal.

4. When the variable is found equal to a case, the statements following that case will execute until a break statement is reached. When a break statement is reached, the switch terminates, and the flow of control jumps to the next line following the switch statement. Not every case needs to contain a break. If no break appears, the flow of control will fall through to subsequent cases until a break is reached.

A switch statement can have an optional default case, which must appear at the end of the switch. The default case can be used for performing a task when none of the cases is true. No break is needed in the default case.

Ask a doubt *(Misuse of 'Ask a Doubt' Section will be dealt as per the Terms & Conditions of Campus Commune)*

Open Doubts | **Closed Doubts**

KARMANJOT

There are lot of topics which are being asked in quizzes but are not covered in the study material. So please tell how to prepare for them.

about 8 days ago

MOHINI GUPTA

The statements before the if block gets executed only if the condition is false
a) TRUE
b) FALSE

What is the answer?

about 22 days ago

SURBHI AGARWAL

false

PRIYANKA SAWANT

The statements above the **if** block have nothing to do with if condition and so ans is FALSE.

PRIYANKA SAWANT

only statemnts inside if are affected by condition

AMARDEEP BURNWAL

none of these

SUMAN SINGH

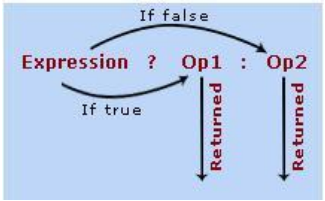
it will always be executed. it has got nothing to do with the if statement

7.2. Ternary Operator and Conditional (Logical) Operators

Java supports another conditional operator that is known as the ternary operator "?:" and basically is used for an if-then-else as shorthand

```
boolean expression ? operand1 : operand2;
```

The "?:" operator evaluates an expression which may also be an operand and returns operand1 if the expression is true; otherwise returns operand2, if the expression is false. We can understand this thing with the help of a diagram shown as:



If we analyze this diagram then we find that, operand1 is returned, if the expression is true; otherwise operand2 is returned in case of false expression.

Lets have an example implementing some Logical operators:

```
class ConditionalOperator {

 public static void main(String[] args){
  int x = 5;
  int y = 10, result=0;
  boolean bl = true;
  if((x == 5) && (x < y))
  System.out.println("value of x is "+x);
  if((x == y) || (y > 1))
  System.out.println("value of y is greater than the value of x");
 result = bl ? x : y;
  System.out.println("The returned value is "+result);
 }
}
```

Output of the Program:

value of x is 5
value of y is greater than the value of x
The returned value is 5

Thus we can conclude the ternary operator can be considered as the short hand of if then else statement.

Conditional (Logical) Operators

Conditional operators return a true or a false value based on the state of the variables i.e. the operations using conditional operators are performed between the two boolean expressions.

Symbol	Name of the Operator
&	AND
&&	Conditional-AND
|	OR
||	Conditional-OR
!	NOT
? :	Ternary (shorthand for if-then-else statement)

I. AND (&) and Conditional-AND (&&) operators:

The AND(&)operator is similar to the Conditional-AND operator (&&).Both of its operands are of boolean type, even these operands may be boolean expressions. Other Relational operators can also be used with these operators.

Lets use a Truth Table to know the status of an output

Op1 or Exp1	Op2 or Exp2	Result
TRUE	TRUE	TRUE
FALSE	FALSE	FALSE
TRUE	FALSE	FALSE
FALSE	TRUE	FALSE

If we analyze the table we find that result is always true only in case of first condition where both operands(or expressions) are true. On the other hand result is always false in other conditions. Note that this table works alike for & and && operators.

In case of "&" operator,if both operands or expressions are true,the result is always true, otherwise it is false if either left-hand or right-hand operand is false.

In case of "&&" operator, the result is also true, if both operands or expressions are true.
But this operator evaluates only the left-hand operand. It doesn't evaluate the right-hand operand if the left-hand operand is false then it will not jump to the right-hand operand to evaluate it, and will come out from that statement and read the next statement. That's why this mechanism is known as short-circuiting. Consider the following statements where the second expression returns false after evaluating only the left-hand operand.

```
True && true = true;// both
operands are evaluated
false && true = false;// only left-
operand is evaluated
```

But the "&" operator always evaluates both of its operands whether the first operand is true or false. Consider the following statements where the second expression returns false after evaluating the right-hand operand.

```
true & true = true;// both
operands are true
true & false = false;// both
operands are evaluated
```

II. OR (|)and Conditional-OR (||)operators:

Likewise, the OR operator(|) is similar to the Conditional-OR operator (||) and returns true, if one or another of its operand is true.
Lets use a Truth Table to know the status of an output that works alike for"|" and "||" operators.

Op1 or Exp1	Op2 or Exp2	Result
TRUE	TRUE	TRUE
FALSE	FALSE	FALSE
TRUE	FALSE	TRUE
FALSE	TRUE	TRUE

If we analyze the table then we find that, result is always false only if both operands or expression are false. On the other hand, result is always true in rest of the other conditions.

Still these exists a major difference in their mode of use:
The "|" operator always evaluates both of its operands and returns true if one or other of its operand is true. Otherwise false if both the conditions are false. Consider the following statements where the second expression returns alse after evaluating the right-hand operand.

```
true | false = true; // left operand is true but
both are evaluated
false | false = false;//both operands are
evaluated
```

In case of "||" the result is also true, if one of the both operands is true. Otherwise it evaluates to falseif both operands are false. But this operator conditionally evaluates the right-hand operand only if the left-hand operands is false. Like the Conditional-AND operator, this mechanism is also known as short-circuiting. Consider the following statements where the first expression returns true after evaluating the right-hand operand.

```
false || true = true;// both operands
are evaluated
true || false = true;// only left-
operand is evaluated
```

III. NOT ("!") operator :

The NOT ("!") operator performs the boolean NOT operation on a single operand or an expression. It checks the boolean status of a current operand or expression and reverses the value of a boolean expression i.e. if the current value of an operand or expression is true then it reverses as

false; but if the value of an operand or expression is false then it reverses as true.

Consider the following example:

```
class BoolNotDemo {
  public static void main(String[] args){
  int x = 2;
  int y = 1;
  boolean bl;

 bl = !(x > y); // bl is false
 System.out.println("x is not greater than y:"+bl);

 bl = !(y > x); // bl is true
 System.out.println("y is not greater than x:"+bl);
  }
}
```

Ask a doubt *(Misuse of 'Ask a Doubt' Section will be dealt as per the Terms & Conditions of Campus Commune)*

7.3. Practice Problems

1. Find the maximum of three numbers
2. Write a program to check whether the input alphabet is a vowel or not.
3. Develop a program, that accepts a deposit amount and calculates amount of interest the deposit amount earns in an year. The bank pays a flat 4% for deposits of up to Rs.1000, a flat 4.5% per year for deposits of up to Rs.5000, and a flat 5% for deposits of more than Rs.5000.
4. Develop the program, which accepts the gross pay and produces the amount of tax owed. For a gross pay of $240 or less, the tax is 0%; for over $. 240 and $. 480 or less, the tax rate is 15%; and for any pay over $480, the tax rate is 28%.
5. Some credit card companies pay back a small portion of the charges a customer makes over a year. A particular credit card company's pay back policy is as follows:
   1. 0.25% for charges up to Rs. 500.
   2. 0.50% for the next Rs.1000 (that is, the portion between Rs. 500 and Rs. 1500),
   3. 0.75% for the next Rs.1000 (that is, the portion between Rs. 1500 and Rs. 2500),
   4. 1.0% for charges above Rs2500.

   Thus, a customer who charges Rs. 400 a year receives Rs.1.00, which is 0.25 • 1/100 • 400, and one who charges Rs1, 400 a year receives

Rs. 5.75, which is 1.25 = 0.25 • 1/100 • 500 for the first Rs. 500 and 0.50 • 1/100 • 900 = 4.50 for the next Rs. 900. Determine by hand the pay backs amount for a customer who charged Rs. 2000 and one who charged Rs. 2600.

Define the program, which accepts a charge amount and computes the corresponding pay back amount.

Click here to download solutions

Ask a doubt *(Misuse of 'Ask a Doubt' Section will be dealt as per the Terms & Conditions of Campus Commune)*

Java

Course Completion Quiz

8.Basic Iterations and Arrays

Objective

- Understand need of arrays in programming
- Understand procedure to declare,initialize and access arrays
- Understand and implement solutions involving arrays of primitive and custom types
- Understand iterations and loops
- Understand and implement solutions involving simple and nested iterations
- Understand finite, infinite loops
- Understand break and continue conditions
- Understand and implement solutions on arrays using iterations

8.3 Finite and Infinite loops

8.4 Break & Continue Keyword

8.5 Practice Problems on arrays using iterations

8.6 Reference Links & videos

Java Lounge

Logout

Java

1.IT Application Overview And Features	Q
2.Need of Programming and Introduction to OOP approach	Q
3.Introduction to Java,JVM,JDK	Q
4.Operators and Variables in Java	Q
5.Introduction to class in java,access specifiers, this and static in Java	Q
6.Java Library, Packages, Use of import	Q
7.Conditional operations	Q
8.Basic Iterations and Arrays	Q
9.Designing Web Pages Part 1(HTML and CSS)	Q
10.Designing Web Pages Part 2 (JavaScript and JQuery)	Q

Course Completion Quiz

8.1. Arrays

We have seen how to declare variable in the previous section. int x; Here x is a variable of type int, so x will be capable of storing an integer value. At any point of time a variable can hold only a single value.

Few more eg.

int age=20;
int score=70;

Say, I need to store scores of last 5 matches. What kind of data type do I take? May be I will take 5 variables of type int.

int scoreMatch1;
int scoreMatch2;
int scoreMatch3;
int scoreMatch4;
int scoreMatch5;

Similarly, I need to record no. of kms I drove each day in last 5 days.

```
int kmsDay1;
int kmsDay2;
int kmsDay3;
int kmsDay4;
int kmsDay5;
```

Similarly, I need to record the amount spent for each day in last 5 days.

```
double expenseDay1;
double expenseDay2;
double expenseDay3;
double expenseDay4;
double expenseDay5;
```

The above approach will not be efficient if the requirement(of recording data for a specific period) changes from just last 5 days to last 30 days.

Array:

It is a single variable, capable of storing multiple values (of same data type). But an array does not store the multiple values directly. It is a collection of more than one variable, of same data type.

Declaring arrays:

The general syntax to declare an array is : data type variable name[]; The square brackets indicate that the variable is an array of the data type declared.

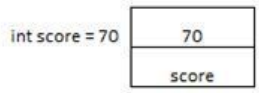
Example:

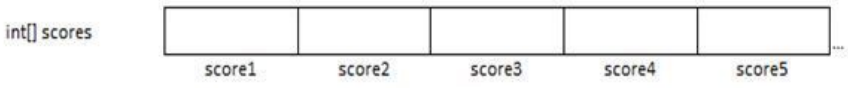
```
int scores [];
int kmsTravelled [];
double expenses [];
char grades[];
```

Also, all the above arrays are just declared, and have not yet been initialized(not defined the size).

Here scores is not just an int type variable. scores is an array of type int i.e. It is capable of storing more than one int value.

int score = 70; Here the variable score holds a value 70.





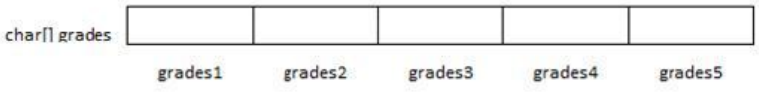
Arrays

Here the array scores do not hold the values directly. It is just a reference (pointing to) to the 5 variables score1 … score5. Hence arrays are called references or objects in Java.

If we do not initialize an int variable, its default value is 0. In case of array (as it does not store value directly), if we do not initialize(do not say to which list of variables it refers to) it, the default value is null. i.e. it does not point to any list of variables.

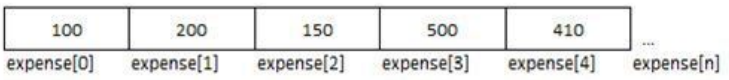
Similarly, expenses is not just a double type variable. expenses is an array of type double i.e. it is capable of storing more than one double value.

Similarly, grades is not just a char type variable. grades is an array of type char i.e. it is capable of storing more than one char value.



Once an array is declared, you can think of multiple variables being generated automatically in a sequential order i.e. if an array double[] expense is

declared, the variable that stores the first value will be expense[0], the variable that stores the second value will be expense[1] and so on.



Note: While talking about arrays, always remember two key things. One, about the array itself, whether it is initialized or not. Two, about the element inside the array – whether it is initialized or not.

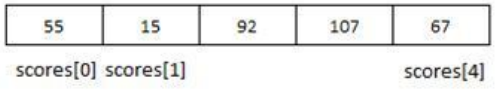
Initializing arrays – using {} initializer:

Arrays have a specific length(or size). The length of the array cannot be changed after it has been initialized. The elements in the array are put at index of the array. The first element in the array is at index 0, and the last element is at the index length-1.

To declare and initialize an array at the same time, refer to the below syntax:

Example 1:

int[] scores = {55,15, 92,107,67}



{} is called an initializer block. The length (or size) of an array is equal to the no. of values declared in the initializer block.

scores is an int array. Its size is 5. The value of first element is 55, value of second element is 15, value of third element is 92, value of fourth element is 107 and value of fifth element is 67.

Example 2:

int[] expenses = {1000,1500,50,200,0}

expenses is an int array. It's size is 5. The value of first element is 1000, value of second element is 1500, value of third element is 50, value of fourth element is 200 and value of fifth element is 0.

Example 3:

String[] softDrinks = {"Coca-Cola", "Pepsi", "Sprite", "Mountain Dew"};

softDrinks is a String array. It's size is 4. The value of first element is Coca-Cola, value of second element is Pepsi, value of third element is Sprite, value of fourth element is Mountain Dew.

All the above examples, declare an array, initialize the array and also insert elements into the array.

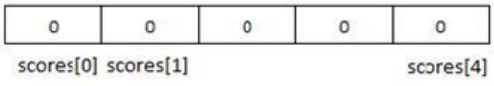
Initializing arrays – using new operator

int scores[];

The above statement just declares an array. It has not been initialized yet i.e. scores is null. Let us initialize the array.

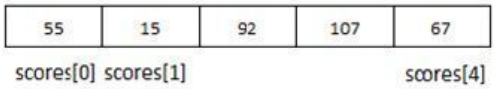
int scores[] = new int[5];

The above statement now initializes the array to size 5. But no elements are put into array. Hence the elements of the array will have their default value. As the array is of type int, all the 5 elements in the array have the value of 0.



You don't need to mention the length(or size) of an array in [] if you are using the initializer block.

int scores[] = new int[]{55,15, 92,107,67};



The above statement declares an array, initializes an array (of size 5) and also puts elements into the array. The value of first element is 55, the value of second element is 15, the value of third element is 92, the value of fourth element is 107 and the value of fifth element is 67.

Once the array is initialized, the length(or size) of an array can be known by the length property. It is used after the dot operator on the array variable. Refer the syntax below:

array.length;

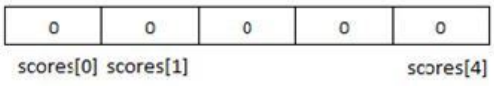
eg.

int scores[] = new int[]{55,15, 92,107,67};
System.out.println(scores.length);

You could either store it in an int variable eg. int size = scores.length;

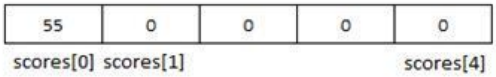
Putting elements into array:

Step 1:
int scores[] = new int[5];

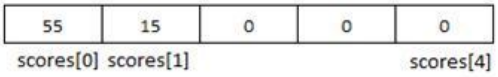


The above statement will declare an int array of length 5. The values in the array will be initialized to the default values as per their data type.
Step 2:
The first value in the array is accessed by the code : scores[0]. To assign values to the first element, use the assignment operator (=). The below statement assigns a value 55 to the first element in the array. The remaining elements still have the default value.

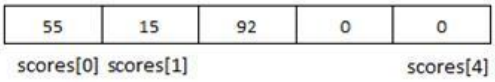
scores[0] = 55;



Similarly, to assign value 15 to the second element.
scores[1] = 15;

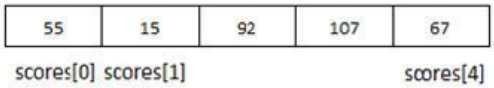


The next line assigns a value 92 to the third element.
scores[2] = 92;



Accessing elements from the array:

int scores[] = new int[]{55,15, 92,107,67};



Assuming that the above array is initialized with the given values. The elements of the array are accessed by their index. The first element of the array is stored at index 0 and the last element is stored at index which is length-1.

To retrieve the value of the elements at index 0, use scores[0] which will return the value 55.

Similarly,

scores[1] will return the value 15
scores[2] will return the value 92
scores[3] will return the value 107
scores[4] will return the value 67

We can use a variable to store the value we retrieve from the array,

int x = scores[0]; as scores is an int array, scores[0] is going to return an int.

Creating array of custom types

Consider the case of library where it has list of books as inventory. So library will have attributes like library name, address. It will also have list of books as an attribute. We could model the Book class and Library class as follow.

```
class Book
{
        int bookId;
        String title;
        String author;
}
```

```
class Library
{
        String libraryName;
        String address;
        Book[] listofBooks;
}
```

The library class has an array of Book, i.e. the array will have multiple objects of Book representing the real books in the library.

> **Creating book objects:**

Assuming the Book class has parameterized constructor.

Book b1 = new Book(101, "Head First Java", "Author1");
Book b2 = new Book(102, "Head First JSP", "Author2");
Book b3 = new Book(103, "Java Complete Reference", "Author3");

> **Creating book array:**

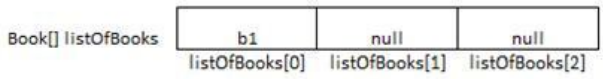
Book[] listOfBooks;

The above statement just declares the array variable. It is not initialized so the listOfBooks is still null. Let us initialize the array.
listOfBooks = new Book[3];

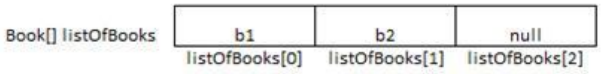
The above statement just initialized the array itself, the elements have not yet been initialized. Let us initialize the elements of the array by assigning the book object value.

> **Putting book objects into the book array**

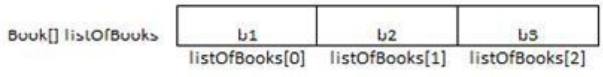
listOfBooks[0] = b1;



listOfBooks[1] = b2;



listOfBooks[2] = b3;



The above 3 statement, initializes all the 3 elements to the value of objects b1, b2 and b3 respectively.

> **Retrieving objects from the book array**

listOfBooks[0] will return the objects b1
listOfBooks[1] will return the objects b2
listOfBooks[2] will return the objects b3

We can use a variable to store the value we retrieve from the array.

Book book = listOfBooks[0]; as listOfBooks is a Book array, listOfBooks[0] is going to return a Book object.

Ask a doubt *(Misuse of 'Ask a Doubt' Section will be dealt as per the Terms & Conditions of Campus Commune)*

Open Doubts **Closed Doubts**

ATUL VISWANATH

taking the example, int[ ] x, y?

Is y an array or not?

about 1 day ago

KARTHIK SAMPATHKUMAR

What are the other ways to sort the elements of array of objects based on its variables without using Comparator and Comparable ??

about 17 days ago

NITISH HANDA

Elements cannot deleted from an array???

about 17 days ago

LALITA KUMARI

When I gave the test,there was one qstn whether arrays can be dynamically created?? I marked it as true and i got the answer wrong.

can anybody help how is it wrong becoz we can create it dynamically and it is known as vector.

about 20 days ago

BHAWNA

hey Lalita

I guess java doesn't support dynamic array.You can go for list and add or remove items from it. So arrays can be dynamically created would be false.

HARIPRASAD R

I guess u may be logically correct as Vector implements array .Even then i guess the class Array can't be instantiated dynamically .

PERABATHINI KIRTI

Java does not support the concept of pointers. That could be the reason for not being able to create them dynamically

SASIVINEETHA GATTUPALLE

Can we initialize arrays like int scores[5]; as in C language? Thank You.

about 22 days ago

PRIYANKA SAWANT

NO,u cannot assign size of array along with its reference.

Bcz the statement int score[]; jst declares an array , array object is not crated until we use new operator.

So we cannot assign a size to areference

PRIYANKA SAWANT

NO,u cannot assign size of array along with its reference.

Bcz the statement int score[]; jst declares an array , array object is not crated until we use new operator.

So we cannot assign a size to areference

PRIYANKA SAWANT

NO,u cannot assign size of array along with its reference.

Bcz the statement int score[]; jst declares an array , array object is not crated until we use new operator.

So we cannot assign a size to areference

PRIYANKA SAWANT

NO,u cannot assign size of array along with its reference.

Bcz the statement int score[]; jst declares an array , array object is not crated until we use new operator.

So we cannot assign a size to areference

PRIYANKA SAWANT

NO,u cannot assign size of array along with its reference.

Bcz the statement int score[]; jst declares an array , array object is not crated until we use new operator.

So we cannot assign a size to areference

PRIYANKA SAWANT

NO,u cannot assign size of array along with its reference.

Bcz the statement int score[]; jst declares an array , array object is not crated until we use new operator.

So we cannot assign a size to areference

PRIYANKA SAWANT

NO,u cannot assign size of array along with its reference.

Bcz the statement int score[]; jst declares an array , array object is not crated until we use new operator.

So we cannot assign a size to areference

PRIYANKA SAWANT

smthng wrng with my browser...multiple ans pasted

SHWETA NAYAK

Is sorting an array easy or difficult?

about 23 days ago

ABUL M

In the above example they had declared as int[] score & char[] grade .Is this a acceptable form to declare a array?

about 2 months ago

DAGGU SAROJA

Yes,this is an acceptable form to declare an array in Java.

This type of declaration is particularly useful in cases when we need to declare multiple arrays of same datatype.

for example,if we want to declare two different arrays: months and days ,both of integer datatype,then instead of declaring them as:

int months[],days[];

we can declare them as follows:

int[] months,days;   which is more efficient.

Hope your doubt is clarified.

SYED MEHNDI

One Dimensional Array

Syntax for default values:

```
int[] num = new int[5];
```

Or (less preferred)

```
int num[] = new int[5]
```

Syntax with values given:

```
int[] num = {1,2,3,4,5};
```

Or (less preferred)

```
int num[] = {1, 2, 3, 4, 5};
```

Note: For convenience int[] num is preferable because it clearly tells that you are talking here about array. Otherwise no difference. Not at all.

Multidimensional array

Declaration

```
int[][] num = new int[5][2];
```

Or

```
int num[][] = new int[5][2];
```

Or

```
int[] num[] = new int[5][2];
```

Java Lounge

Logout

Java

Course Completion Quiz

8.2. Loops and Iterations

Let us take a scenario where a teacher has to perform task of giving grades to the students according to their percentage. He/she checks the percentage of first student and enters the grade, then checks the percentage of second student and enters the grade, same procedure for the next student as well. Here, the teacher has to perform the same task again and again. Don't you think this is a laborious and error prone work? Yes it is.

One of the things computers are often used for is the automation of repetitive tasks. Repeating identical or similar tasks without making errors is something that computers do well and humans do poorly. In this chapter, we shall study in detail how the repeated tasks can be automated using Java programming.

Let us see a simple java program to print first 10 natural numbers.

Program:

```
public class Sample
	{
		public static void main(String args[])
		{
			int n=1;
			System.out.print(n++);
			System.out.print(n++);
			System.out.print(n++);
			System.out.print(n++);
			System.out.print(n++);
			System.out.print(n++);
			System.out.print(n++);
			System.out.print(n++);
			System.out.print(n++);
			System.out.print(n++);
		}
	}
```

Output:

12345678910

In the above program, we wrote the same line of code 10 times to obtain output. This is nothing but code redundancy(repetition) which makes our program an inefficient one. To eliminate this we use loops and iterations in Java.

Loop is a block of code executed repeatedly until some condition is satisfied and one pass through(execution of) the loop is called iteration.

Here condition is a boolean expression. A boolean expression is an expression written using conditional operators(<, >, ==, <=,>= etc) and the result of it will always be either true or false.

Loops in Java are mainly of three types :-

1. 'while' loop
2. 'do while' loop
3. 'for' loop

While loop:

A while loop is a control structure that allows you to repeat a task certain number of times. In simple words, using while loop we can execute a set of statements several times based on a condition.

> **Syntax:**
>
> ```
> while(condition)
> {
>  //Statements
> }
> ```

Here, "while" is a keyword and "condition" is a boolean expression and as long as this condition yields true, the statements within the braces({}) are executed. You can almost read a while statement as if it were English- "while condition is true, continue executing the Statements".

Let us now write the same program(print first 10 natural numbers) using while loop.

Program:

```
public class WhileExample
{
	public static void main(String args[])
	{
		int n=1;
		while(n<=10)
		{
		System.out.print(n++);
		}
	}
}
```

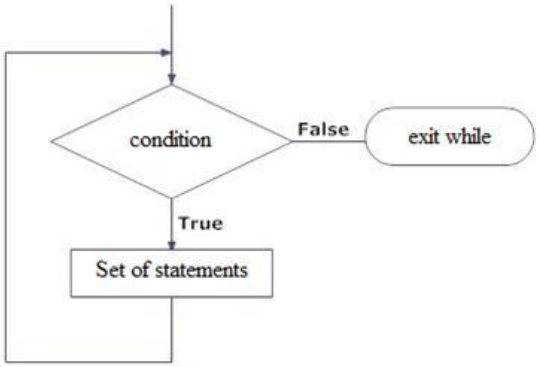
Output:

12345678910

In the above code snippet, we eliminated those 9 repeated lines, clubbed them into a single line and obtained the output using while. Here n

is also called counter variable and it is used to iterate through the loop.

Flow of control(while loop):



Flowchart of while loop

The flow of execution for a while statement is as follows:

1. Evaluate the condition, whether the result is true or false.
2. If the result is false, exit the while statement and and execute the first statement after the while loop.
3. If the result is true, execute the set of statements within the curly braces, and then go back to step 1.

Example:

/*Program to automate the grade generation of five students based on their percentage using while loop */

```
public class GradeGenerationUsingWhile {
  public static void main(String args[])
  {
      /*initialize an array of integer type that stores percentage of students
       * assuming that they are in sequential order of their roll numbers*/
      int[] percentage=new int[]{50,46,32,98,75};
      //initialize a variable that is used to iterate through the percentage array
      int rollNo=0; //initialized to 0 because array index starts from 0

      while(rollNo<percentage.length)//until rollNo reaches the end of array
      {
          System.out.print("Roll no "+(rollNo+1)+": "); //to print rollNo
          if(percentage[rollNo]<40) //if percentage is below 40
          {
              System.out.println("Grade is C");
          }

          //if percentage is between 40-70
          else if(percentage[rollNo]>40&&percentage[rollNo]<70)
          {
              System.out.println("Grade is B");
          }
          else //if percentage is above 70
          {
              System.out.println("Grade is A");
          }
          rollNo++; //next rollNo
      }
  }
}
```

Output:

Roll no 1: Grade is B

Roll no 2: Grade is B

Roll no 3: Grade is C

Roll no 4: Grade is A

Roll no 5: Grade is A

Flow of the above program:

1. Initialize an integer array of percentage.
2. Initialize an integer variable rollNo=0; which is used to iterate through the array.

3. Since we need to generate grades for only those students whose percentage is given, we are specifying the condition in while as

rollNo<percentage.length. We have learned in the previous topic that .length property gives the size of array which is 4 in the above program. rollNo points to the index of the array. So before entering while loop, the value of rollNo is checked.
4. If the condition is satisfied(if rollNo<percentage.length) go to step 5 else go to step 6.
5. Now control is sent to the body of loop

- In the body of while loop, if-else conditions are specified to check the percentage of student marks. Based on those conditions respective grade is printed.
- Increment rollNo.
- Go to Step 4.

6. End.

Do... While loop:

It is very similar to the 'while' loop shown above. The only difference being, in a 'while' loop, the condition is checked beforehand, but in a 'do... while' loop, the condition is checked after one execution of the loop. Hence, the body of do... while loop is executed at least once.

Syntax:

```
do
{
  //statements
}while(condition);
```

Here, "while" and "do" are keywords and "condition" is a boolean expression. Each iteration of the do-while loop first executes the statements and then evaluates the condition. If the result of this condition is true, the loop will repeat. Otherwise, the loop terminates. Notice, there is semicolon at the end of while(); in do... while loop.
Let us now write the same program(print first 10 natural numbers) using do... while loop.
Program:

```
public class doWhileExample
{
    public static void main(String args[])
    {
        int n=0;
        do
        {
          System.out.print(n);
        }while(n<=10);
    }
}
```

Output:
12345678910

Now you may get a question as why "do.. while" when we already have "while" and when both are giving the same output? We shall study about it in detail.

Why do... while and how it differs from while?

If the condition controlling a while loop is initially false, then the body of the loop will not be executed at all. However, sometimes it is desirable to execute the body of a while loop at least once, even if the conditional expression is false to begin with. In other words, there are times when you would like to test the condition at the end of the loop rather than at the beginning.

Fortunately, Java supplies a loop that does just that: the do-while. The do-while loop always executes its body at least once, because its condition is at the bottom of the loop. Therefore, while is called entry-called loop whereas do...while is called exit-controlled loop.
The difference is best illustrated using the programs below:

/*Program to demonstrate difference between while and do..while*/

Do-while example:

```
public class doWhileDemo
{
      public static void main(String args[])
      {
            int x=11;
            do
            {

               System.out.print("do-while: "+x++);

            }while(x<=10);
      }
}
```

Output:

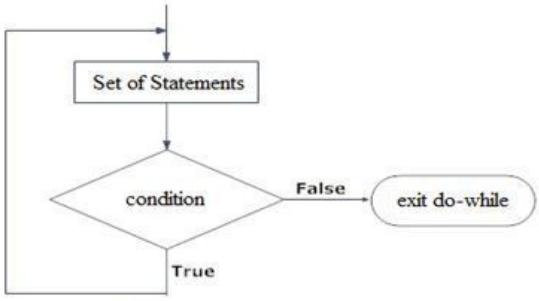
do-while: 11

While example:

```
public class WhileDemo
{
      public static void main(String args[])
      {
            int x=11;
            while(x<=10)
            {

               System.out.print("while: "+x++);

            }
      }
}
```

Output:(no output)

Flow of control(do...while loop):



The flow of execution for a do... while statement is as follows:

1. Execute the set of statements.
2. Evaluate the condition, whether the result is true or false.
3. If the result is false, exit the while statement and execute the first statement after the do... while loop.
4. If the result is true, go back to step 1.

Example:

/*Program to generate the grade for five students of a class based on their percentage using do... while loop */

```
public class GradeGenerationUsingDoWhile{
 public static void main(String args[])
 {
   /*initialize an array of integer type that stores percentage of students
   assuming that they are in sequential order of roll numbers*/
   int[] percentage=new int[]{50,46,32,98,75};

   //initialize a variable that is used to iterate through the percentage array
   int rollNo=0; //initialized to 0 because array index starts from 0

   do //execute the following statements
   {
       System.out.print("Roll no "+(rollNo+1)+": "); //to print rollNo
       if(percentage[rollNo]<40) //if percentage is below 40
       {
           System.out.println("Grade is C");
       }
       else if(percentage[rollNo]>40&&percentage[rollNo]<70)//if percentage is between 40-70
       {
           System.out.println("Grade is B");
       }
       else //if percentage is above 70
       {
           System.out.println("Grade is A");
       }
       rollNo++; //increment rollNo
   }while(rollNo<percentage.length);//as long as this condition yields true go for next iteration
  }
}
```

Output:

Roll no 1: Grade is B

Roll no 2: Grade is B

Roll no 3: Grade is C

Roll no 4: Grade is A

Roll no 5: Grade is A

Flow of the above program:

1. Initialize an integer array of percentage.
2. Initialize an integer variable rollNo=0; which is used to iterate through the array
3. Execute step 4.
4. Now control is sent to the body of loop

i. In the body of do... while loop, if-else conditions are specified to check the percentage of student marks. Based on those conditions respective grade is printed.

ii. Increment rollNo.

5. Since we need to generate grades for only those students whose percentage is given, we are specifying the condition in while as rollNo<percentage.length. We have learned in the previous topic that .length property gives the size of array which is 4 in the above program. So before proceeding with the next iteration, the value of rollNo is checked.
6. If the condition is satisfied(if rollNo<percentage.length) go to step 3 else go to step 7.
7. End

For loop:

A for loop is a repetition control structure that allows you to efficiently write a loop that needs to execute a specific number of times. A for loop is useful when you know how many times a task is to be repeated.

Syntax:

```
for(initialization; condition; increment/decrement)
{
  //Statements
}
```

Here,the initialization step is executed first, and only once. This step allows you to declare and initialize any loop control variables. Next, the condition is evaluated. If it is true, the statements are executed, else control comes out of for loop and the next statement after the loop is executed.

Increment/decrement is an update statement which gets executed after the execution of body of loop.

Let us now write the same program(print first 10 natural numbers) using for loop.

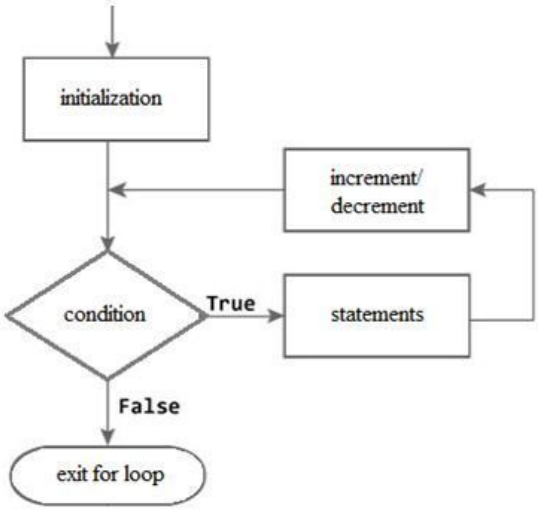
Program:

```
public class ForExample
{
      public static void main(String args[])
      {
            for(int n=1;n<=10;n++)
            {
            System.out.print(n);
            }
      }
}
```

Output:

12345678910

Flow of control(for loop):



The flow of execution for a while statement is as follows:

1. Initialize the variable. Initialization step is executed first, and only once.
2. Evaluate the condition, whether the result is true or false.
3. If the result is false, exit the for loop and and execute the first statement after the for loop.
4. If the result is true, execute the set of statements within the curly braces, and then go to increment/decrement statement.
5. Go to step 2.

Example:

/*Program to generate the grades for five students based on their percentage using for loop */

```
public class GradeGeneration{
   public static void main(String args[])
   {
       /*initialize an array of integer type that stores percentage of students
       assuming that they are in sequential order of roll numbers*/
       int[] percentage=new int[]{50,46,32,98,75};

       //initialize a variable that is used to iterate through the percentage array
       int rollNo;

       for(rollNo=0;rollNo<percentage.length;rollNo++)//execute the following as long as condition is satisfied
       {
           System.out.print("Roll no "+(rollNo+1)+": "); //to print rollNo
           if(percentage[rollNo]<40) //if percentage is below 40
           {
               System.out.println("Grade is C");
           }
           else if(percentage[rollNo]>40&&percentage[rollNo]<70)//if percentage is between 40-70
           {
               System.out.println("Grade is B");
           }
           else //if percentage is above 70
           {
               System.out.println("Grade is A");
           }

       }

   }
}
```

Output:

Roll no 1: Grade is B

Roll no 2: Grade is B

Roll no 3: Grade is C

Roll no 4: Grade is A

Roll no 5: Grade is A

Flow of the above program:

1. Initialize an integer array of percentage.
2. Declare an integer variable rollNo; which is used to iterate through the array
3. Initial value of rollNo is 0 and this statement is executed only once in the entire cycle of for loop.
4. Since we need to generate grades for only those students whose percentage is given, we are specifying the condition as rollNo<percentage.length. We have learned in the previous topic that .length property gives the size of array which is 4 in the above program. So before entering into the for loop, the value of rollNo is checked.
5. If the condition is satisfied(if rollNo<percentage.length) go to step 6 else go to step 7.
6. Now control is sent to the body of loop

i. In the body of while loop, if-else conditions are specified to check the percentage of numbers. Based on those conditions respective grade is printed.

ii. Increment rollNo.

iii. Go to Step 5.

7. End.

Advanced for loop:

This is an enhanced for loop which mainly used for arrays. Advanced for loop is also called 'for each' loop.

Syntax:

```
for(declaration : expression)
{
  //Statements
}
```

Declaration: The newly declared block variable, which is of a type compatible with the elements of the array you are accessing. The variable will be available within the for block and its value would be the same as the current array element.

Expression: This evaluates to the array you need to loop through. The expression can be an array variable or method call that returns an array.

Let us take the same example of grade generation and see how it works in advanced for loop.

Example:

/*Program to generate the grades of 5 students based on their percentage using advanced for loop */

```
public class GradeGeneration{
   public static void main(String args[])
   {
       //initialize an array of integer type that stores percentage of students
       int[] percentage=new int[]{50,46,32,98,75};

       for(int eachPercentage:percentage)//execute the following until eachPercentage reaches end of array
       {
           System.out.print("Percentage is "+(eachPercentage)+": "); //to print eachPercentage
           if(eachPercentage<40) //if percentage is below 40
           {
               System.out.println("Grade is C");
           }
           else if(eachPercentage>40&&eachPercentage<70)//if percentage is between 40-70
           {
               System.out.println("Grade is B");
           }
           else //if percentage is above 70
           {
               System.out.println("Grade is A");
           }

       }

   }
}
```

Output:

Percentage is 50: Grade is B
Percentage is 46: Grade is B
Percentage is 32: Grade is C
Percentage is 98: Grade is A
Percentage is 75: Grade is A

In the above program, eachPercentage is a variable that is of the same data type as array of percentage. In the previous program rollNo takes the index value whereas here eachPercentage takes value at index. Therefore, initially the value of eachPercentage is percentage[0] which is 50 and in the second iteration eachPercentage will take the value percentage[1] which is 46 and so on.

Nested loops:

We have seen the advantages of using various methods of iteration, or looping. Now let's take a look at what happens when we combine looping procedures. The placing of one loop inside the body of another loop is called nesting.

When you "nest" two loops, the outer loop takes control of the number of complete repetitions of the inner loop. While all types of loops may be nested, the most commonly nested loops are for loops.

Let us look at an example on nested loops:

/*Program to print the following pattern
1
12
123
1234
12345
*/

```
public class NestedLoop
{
public static void main(String args[])
  {
      for(int i=1;i<=5;i++) //outer loop
      {
            for(int j=1;j<=i;j++) //inner loop
            {
                  System.out.print(j);
            }
            System.out.println();
      }
  }
}
```

Output:

1

12

123

1234

12345

In the above program, outer loop is used to determine the no.of digits to be printed in a line whereas the inner loop is used to print the digits. We have used print() within the body of inner loop because we want the digits to be displayed in the same line and after the inner loop is executed, we go to the next line using println() statement.

Ask a doubt *(Misuse of 'Ask a Doubt' Section will be dealt as per the Terms & Conditions of Campus Commune)*

Open Doubts | **Closed Doubts**

SHANKAR PAUL

The program is correct because n is incremented(n++)

about 1 month ago

SHANKAR PAUL

In the example program for do while loop there is no incrementation of the do while loop but still it is printing the list of numbers as given.How is it possible?

about 2 months ago

SRAVYA SHIVAPURAM

The output given is wrong! The program produces infinite loop as the value of n is not incremented. The output is infinite number of 0's as the condition is always true (i.e 0<=10).

ANAND GUPTA

I guess they forgot to add an incremental statement while editing from another source

SAGAR JAIN

No its right, Check out the last statement of "do" loop.
Its the increment roll no++ working there.

8.3. Finite and Infinite loops

A finite loop ends itself i.e control comes out of the loop at certain point of time. For example, the for loop in the previous program that displays multiples of 3 is a finite loop because the loop exits if n>10.

In other case,a loop becomes an infinite loop if a condition never becomes false. The for loop is traditionally used for this purpose. Since none of the three expressions that form the for loop are required, you can make an endless loop by leaving the conditional expression empty.

Example for infinite loop:

```
public class InfiniteLoop
{
   public static void main(String args[])
   {
     for(;;)
     {
     System.out.println("Tata Consultancy Services");
     }

   }
}
```

Output:

Tata Consultancy Services
Tata Consultancy Services
Tata Consultancy Services
.
.
.
.

This goes on printing until we forcefully stop the execution. The above program is an example of implementing infinite loop using for loop. Similarly, you can implement an infinite loop using while and do... while as follows.

Infinite loop using while:

```
while(true)
{
   // your code goes here
}
```

Infinite loop using do... while:

```
do{
   // your code goes here
} while(true);
```

Ask a doubt *(Misuse of 'Ask a Doubt' Section will be dealt as per the Terms & Conditions of Campus Commune)*

8.4. Break & Continue Keyword

The break keyword is used to stop the entire loop. Using break we can leave a loop even if the condition for its end is not fulfilled. It can be used to end an infinite loop, or to force it to end before its natural end.

Syntax:

break;

Example: Write a program to print a list of numbers until 3 is encountered

Program:

```
public class BreakExample
{
   public static void main(String args[])
   {
     for (int x = 1; x < 6; x++)
     {
         if(x==3)
         {
               System.out.println("Break the loop");
               break;
         }
         System.out.println("x is " + x);
     }
     System.out.println("Exit");
   }
}
```

Output:

x is 1

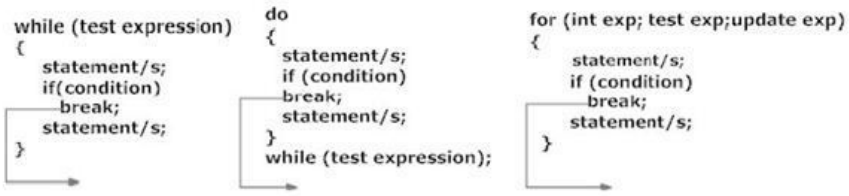
x is 2

Break the loop

Exit

In the above example if the value of x is 3 then the control enters into the loop and when break statement is encountered, it comes out of the for loop and executes the first statement after the for loop. The break statement can be used in terminating all three loops for, while and do...while.

Working of break statement:



The continue keyword can be used in any of the loop control structures. Continue statement skips the current iteration of a for, while or do while loop. It causes the loop to immediately jump to the next iteration of the loop.

In a for loop, the continue keyword causes flow of control to immediately jump to the increment/decrement statement.

In a while loop or do/while loop, flow of control immediately jumps to the condition.

Syntax:

continue;

Example: Write a program to print numbers between 1 to 5 except 3.

```
public class ContinueExample
{
   public static void main(String args[])
   {
     for (int x = 1; x < 6; x++)
     {
         if(x==3)
         {
               System.out.println("Continue the loop");
               continue;
         }
         System.out.println("x is " + x);
     }
     System.out.println("Exit");
   }
}
```

Output:

x is 1

x is 2

Continue the loop

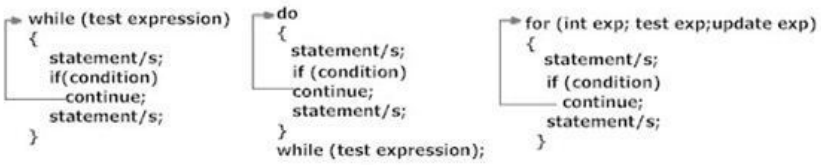
x is 4

x is 5

Exit

In the above example if the value of x is 3 then the control enters into the loop and when continue statement is encountered, it jumps to the next iteration without executing the rest of the statements within the loop.

Working of continue statement:



Ask a doubt *(Misuse of 'Ask a Doubt' Section will be dealt as per the Terms & Conditions of Campus Commune)*

Open Doubts | **Closed Doubts**

GAURAV GUPTA

If continue; statement is before increment/decrement statement in while or do-while loop then loop will become infinite or it will terminate...?

about 29 days ago

PRIYANKA SAWANT

infinite

DAGGU SAROJA

Whenever continue is encountered in do-while loop,the control should jump to the test expression.But in the figure shown above for the working of continue statement in do-while loop,it is shown that the control jumps to the starting of the loop.Is that right?

about 2 months ago

LAKSHMINADHA PRASAD THOTA

Dear Saroja..

You are correct...in do-while case after **continue** the control should jump to test expression. If test expression True then only it will execute do statements.

TATA CONSULTANCY SERVICES

Java Lounge

Logout

Java

1.IT Application Overview And Features	Q
2.Need of Programming and Introduction to OOP approach	Q
3.Introduction to Java,JVM,JDK	Q
4.Operators and Variables in Java	Q
5.Introduction to class in java,access specifiers, this and static in Java	Q
6.Java Library, Packages, Use of import	Q
7.Conditional operations	Q
8.Basic Iterations and Arrays	Q
9.Designing Web Pages Part 1(HTML and CSS)	Q
10.Designing Web Pages Part 2 (JavaScript and JQuery)	Q

Course Completion Quiz

8.5. Practice Problems on arrays using iterations

1./*Program to search for an element in the given array*/

In the program below we are searching for 90.

```
public class Search
{
 public static void main(String args[])
 {

  int numbers[] = new int[]{55,15,92,107,67}; //initializing an array
  boolean isFound=false;
  int i=0;


  for(i=0;i<numbers.length;i++) //iterating through array
  {
    if(numbers[i]==90) //if number is present do the following
    {
      isFound=true;
      break;               //break the loop
    }
  }//end for


  if(isFound==true)
  {
    System.out.println("90 is found at position: "+(i));
  }
  else
  {
    System.out.println("Element is not present in the given set of numbers");
  }

 }
}
```

Output: Element is not present in the given set of numbers.

2./*Program to copy elements from one array to the other using for loop*/

```
public class CopyArray
{
  public static void main(String args[])
    {
          int[] array1={11,12,13,14,15};
          int[] array2=new int[10];

          System.out.println(" The contents of the Array-1 are :");
          for(int j=0;j<array1.length;j++)
          {
            System.out.println(" " + array1[j]);
          }
          //copying elements from array1 to array2 using for loop
          for(int i=0;i<array2.length;i++)
          {
              if(i<array1.length)
                  array2[i]=array1[i];
              else
                  array2[i]=array2[i-1]+1;
          }

          System.out.println("\n The contents of the Array-2 are :");
          for(int l=0;l<array2.length;l++)
          {
              System.out.println(" " + array2[l]);
          }

    }
  }
```

Output:

```
The contents of the Array-1 are :
11
12
13
14
15

The contents of the Array-2 are :
11
12
13
14
15
16
17
18
19
20
```

3. Practice problems on arrays using iterations:

/*Program to find the costliest book in a library*/

```
public class Library {

 //defining the attributes of Library
 String libraryName;
 String address;
 Book[] listOfBooks;

 //parameterized constructor
 public Library(String libraryName, String address, Book[] listOfBooks) {

  this.libraryName = libraryName;
  this.address = address;
  this.listOfBooks = listOfBooks;
 }

 //method to find the costliest among all the books present in library
 public Book getCostliestBook()
 {
   Book costliestBook=listOfBooks[0];//assuming that first book is costliest book
   for(int i=0;i<listOfBooks.length;i++)//iterating through array of Books
   {
      //comparing the price of one book with that of the other
     if(listOfBooks[i].price>costliestBook.price)// checking if the book is
                                                 costlier than the previous one
     {
       costliestBook=listOfBooks[i];//costliestBook holds a new value
     }
   }
  return costliestBook; //return costliest book
 }

  public static void main(String args[])
  {
   Book[] listOfBooks=new Book[3];//declaring array of Books
   //creating book objects
   Book b1 = new Book("Head First Java",550.0);
   Book b2 = new Book("Head First JSP", 980.0);
   Book b3 = new Book("Java Complete Reference", 500.0);
   //putting book objects into the book array
   listOfBooks[0]=b1;
   listOfBooks[1]=b2;
   listOfBooks[2]=b3;
   //creating library object
   Library library=new Library("Universal","Gandhinagar",listOfBooks);
  //calling getCostliestBook() method which returns an object of type Book
   Book costliestBook=library.getCostliestBook();
   //print the details of costliest book
   System.out.println("Costliest Book in the library is: "+costliestBook .title+"
                 with cost of rupees "+costliestBook .price);
  }
 }
```

```
public class Book {

    //define attributes for Book class
    String title;
    double price;

    //parameterized constructor
    public Book(String title, double price) {
        this.title=title;
        this.price=price;
    }
}
```

Output:Costliest Book in the Library is: Head First JSP with cost of rupees 980.0

Points to remember:

Arrays

- Arrays are objects(not primitive) in Java.
- You create an object of array either by using the initializer block {} or by using the new operator.
- Arrays have a constant length, i.e. the length cannot increase dynamically at run time.
- The first index of an array is 0 and the last index is length-1. eg. if the array declared is of size 4, the values are stored at index 0 to 3.
- The property length is used to find the length(or size) of an array.
- If the elements in an array are not assigned any value, the value will be the default value depending on the type of array. eg. int[] will have value 0 at index where the value is not initialized, String[] will have value null at index where the value is not initialized.

Iterations

- Iterations are used to execute repetitive tasks.
- Condition in the while,do... while and for should give result as false after several iterations to terminate the loop.
- A loop control variable used to count the iterations is called counter variable.

- Do not forget to update the counter variable. Ex:n++,i-- etc.
- In do while loop, there is a semicolon after while statement.
- In for each loop, declared variable and array should be of same data type.
- The break statement would stop the execution.
- The continue statement skips the current iteration.

Ask a doubt *(Misuse of 'Ask a Doubt' Section will be dealt as per the Terms & Conditions of Campus Commune)*

TATA CONSULTANCY SERVICES

Java Lounge

Logout

Java

1.IT Application Overview And Features	Q
2.Need of Programming and Introduction to OOP approach	Q
3.Introduction to Java,JVM,JDK	Q
4.Operators and Variables in Java	Q
5.Introduction to class in java,access specifiers, this and static in Java	Q
6.Java Library, Packages, Use of import	Q
7.Conditional operations	Q
8.Basic Iterations and Arrays	Q
9.Designing Web Pages Part 1(HTML and CSS)	Q
10.Designing Web Pages Part 2 (JavaScript and JQuery)	Q

Course Completion Quiz

9.Designing Web Pages Part 1(HTML and CSS)

Objectives

- To get introduced with Hyper Text Markup Language (HTML)
- To get introduced with Cascading Style Sheet (CSS).

9.1 HTML Elements

9.2 HTML Links, Images, Division, List

9.3 HTML Table

9.1. HTML Elements

What is HTML?

To understand HTML, let us first see the meaning of a Markup language

A markup language is a modern system for annotating a document in a way that is syntactically distinguishable from the text.

See example as indicated below:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE recipe PUBLIC "-//Happy-Monkey//DTD RecipeBook//EN"
"http://www.happy-monkey.net/recipebook/recipebook.dtd">

<recipe>

    <title>Peanut-butter On A Spoon</title>

    <ingredientlist>
        <ingredient>Peanut-butter</ingredient>
    </ingredientlist>

    <preparation>
        Stick a spoon in a jar of peanut-butter,
        scoop and pull out a big glob of peanut-butter.
    </preparation>

</recipe>
```

The blue text shown in the example is the Markups to indicate the content enclosed in it.
HTML or HyperText Markup Language is the main markup language for creating web pages and other information that can be displayed in a web browser. It is a markup language that web browsers use to interpret and compose text, images and other material into visual or audible web pages.

Now it is simple. Isn't it? We may summarize the knowledge of HTML as below

- HTML stands for Hyper Text Markup Language
- HTML is a markup language
- A markup language is a set of markup tags
- The tags describe document content
- HTML documents contain HTML tags and plain text
- HTML documents are also called web pages

A HTML document when composed of the markup elements is saved as a file with a .html or .htm extension. The rendering of the document is done on any web browser such as Internet Explorer, Google Chrome, etc.,.

HTML Elements

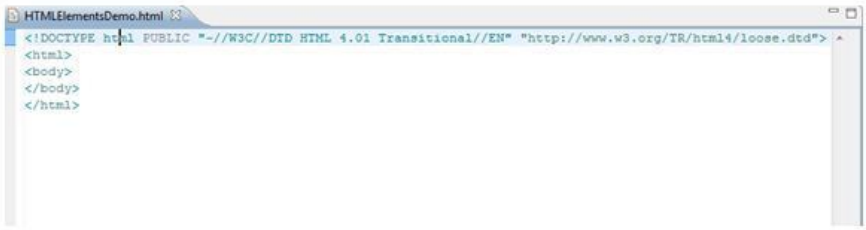
HTML is written in the form of HTML elements consisting of tags enclosed in angle brackets (like <html>), within the web page content. HTML tags most commonly come in pairs like <h1> and </h1>, although some tags represent empty elements and so are unpaired, for example <img>. The first tag in a pair is the start tag, and the second tag is the end tag (they are also called opening tags and closing tags). In between these tags web designers can add text, further tags, comments and other types of text-based content. Most HTML elements have attributes.

The browser does not display the HTML tags, but uses the tags to interpret the content of the page. The purpose of a web browser is to read HTML documents and compose them into visible web pages.
Let us see each of the HTML element in detail:

DOCTYPE

The <!DOCTYPE> declaration helps the browser to display a web page correctly.

There are many different documents on the web, and a browser can only display an HTML page 100% correctly if it knows the HTML type and version used
Below is a HTML document showing the <!DOCTYPE> declaration in the first line.



HTML

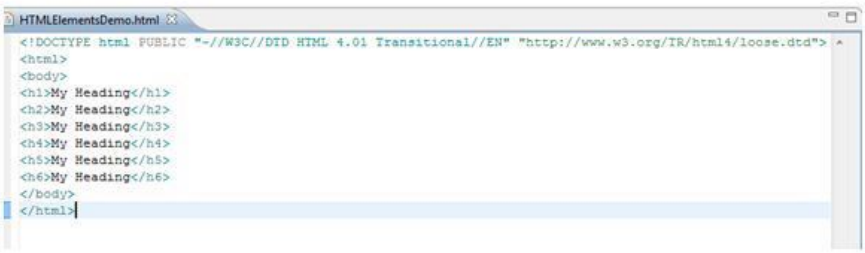
The <html> element defines the whole HTML document. The element has a start tag <html> and an end tag </html>.

BODY

The <body> element defines the body of the HTML document. The element has a start tag <body> and an end tag </body>.

Headings

HTML headings are defined with the <h1> to <h6> tags



Output of the above code may be viewed using a web browser as below:

My Heading

My Heading

My Heading

My Heading

My Heading

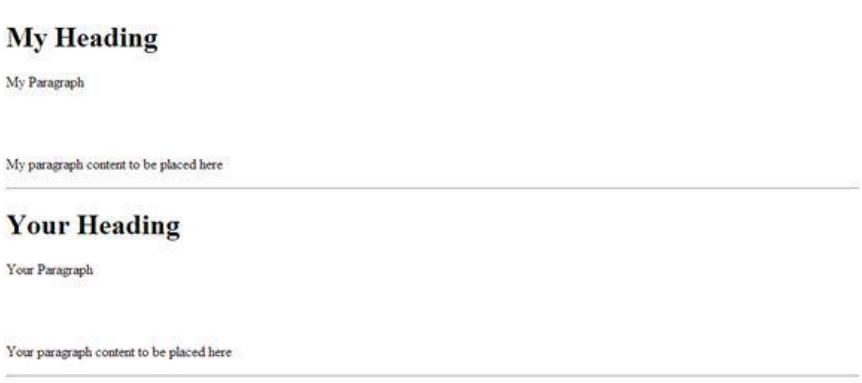
My Heading

Various other elements in HTML

The other basic elements that are widely used in HTML are tabulated as below

Element	Description
<p>	Defines a paragraph in the HTML document
 	Defines a line break
<hr>	Defines a horizontal line
<!-->	Defines a comment in the HTML document and is not interpreted by browser

Sample code and its output:



HTML Head elements

The elements of the head element with the associated nested tags are listed in the table below

Element	Description
<head>	Defines information about the document
<title>	Defines the title of a document
<style>	Defines style information for a document
<link>	Defines the relationship between a document and an external resource
<meta>	Defines metadata about an HTML document
<script>	Defines a client-side script

The sample HTML document consisting of these head elements is shown here with associated output screen

HTML formatting elements

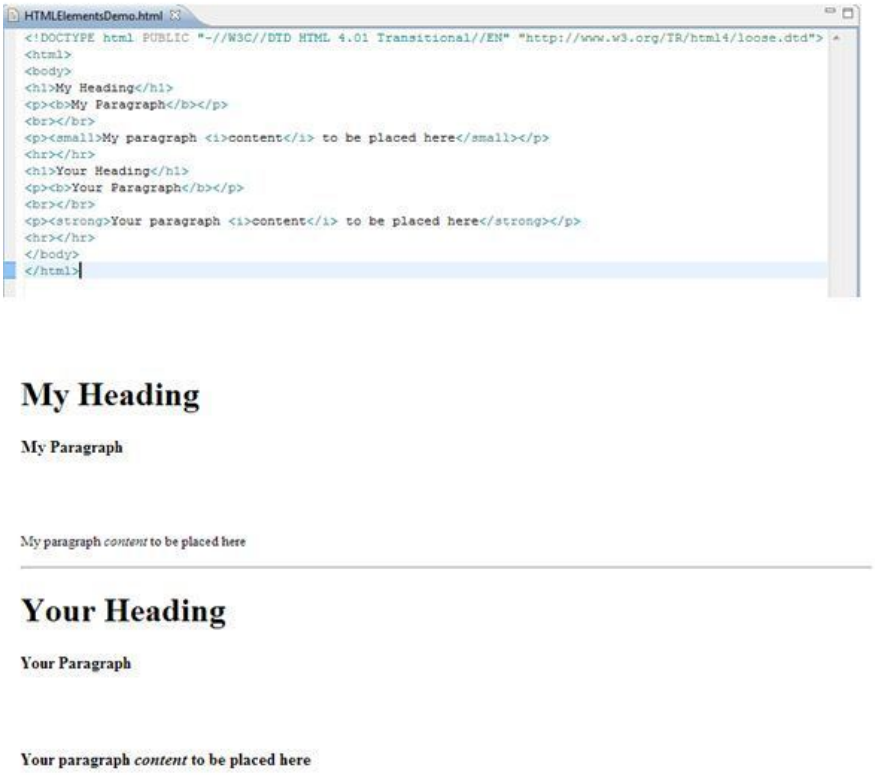
The HTML formatting elements have been tabulated here.

Element	Description
<b>	Defines bold text
<i>	Defines in italics
<small>	Defines smaller text
	Defines important text
<sub>	Defines subscripted text
<sup>	Defines superscripted text

Sample code for some of the formatting tags are given below. The output has also been displayed along with.

HTMLElementsDemo.html

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
<html>
<body>
<h1>My Heading</h1>
<p><b>My Paragraph</b></p>
<br></br>
<p><small>My paragraph <i>content</i> to be placed here</small></p>
<hr></hr>
<h1>Your Heading</h1>
<p><b>Your Paragraph</b></p>
<br></br>
<p><strong>Your paragraph <i>content</i> to be placed here</strong></p>
<hr></hr>
</body>
</html>
```



HTML Attributes

Attributes in HTML element provide additional information about the element.

Attributes	Description
class	Specifies one or more classnames for an element (refers to a class in a style sheet)
id	Specifies a unique id for an element
style	Specifies an inline CSS style for an element
title	Specifies extra information about an element (displayed as a tool tip)

Ask a doubt *(Misuse of 'Ask a Doubt' Section will be dealt as per the Terms & Conditions of Campus Commune)*

Open Doubts | **Closed Doubts**

TANAY DAS

need help plz:

1.adding a border to an image help to identify link??frame?? or beauty of the image??? {frame-wrong}

2.<td>...</td> for inserting steps?? column?? row?? table??? {row-wrong}

3.Identify the invalid tag? a) <fontface='verdana'> b) <fontface='verdana,arial'> c) <font_face='verdana'> d) <font='verdana'> (d-wrong)

4.adding alternative text to images will help user to save image in this name// or to get idea before it loads//or help user when they wishes to insrt another picture (dnt knw)

about 14 days ago

SUMAN SINGH

1) just for the beauty and nothing else

2)

<td> is used to enter a data field in the tr.

for example in the last table of this page - the html attribute u would use

<tr>

<td>class</td>

<td> Specifies one or more.... </td>

</tr>

In short tr will be used to enter a row and td will be used to enter a data in that row. each td would enter a value in a different column

3. question not clear. you have already mentioned the wrong answer

4. it will help in the SEO of the image.. it will be displayed when the image is not displayed

SAYANTAN GANGOPADHYAY

1)width of border around image

2)table

3)b

4)get idea before it loads

SAYANTAN GANGOPADHYAY

sorry the ans of ques 2 is column

SHYAM VASANI

When the IRA will be conducted? Will it be on the first day of joining or there will be gap of few days?

about 15 days ago

ATUL VISWANATH

On the second day of ILP.

9.2. HTML Links, Images, Division, List

HTML Links

The HTML <a> tag defines a hyperlink.

A hyperlink (or link) is a word, group of words, or image that you can click on to navigate to another document.

When you move the cursor over a link in a Web page, the arrow will turn into a little hand. The most important attribute of the <a> element is the href attribute, which indicates the link's destination.

By default, links will appear as follows in all browsers:

- An unvisited link is underlined and blue
- A visited link is underlined and purple
- An active link is underlined and red

Samples for defining HTML link are shown here:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta name="keywords" content="HTML, headTag, title, link, script">
<title>HTML Head Demo</title>
<link rel="stylesheet" type="text/css" href="mystyle.css">
<script type="text/javascript">
</script>
<style type="text/css">
body {background-color:white;}
p {color:black;}
</style>
</head>
<body>
<h4><a href="http://www.google.com/">Search</a>
</h4>
<p><a id="paraId">My Paragraph<br/>
Lorem ipsum dolor sit amet, consectetur adipiscing elit
</a></p>
</body>
</html>
```

Search

My Paragraph
Lorem ipsum dolor sit amet, consectetur adipiscing elit

HTML Images

In HTML, images are defined with the <img> tag. The <img> tag is empty, which means that it contains attributes only, and has no closing tag. Some of the attributes of the image tag are

- src – Indicates the source or the URL of the image that is to be displayed
- alt – alternate name or text for the image
- height, width – attributes describing the size

Below is an example (image will be visible if it is placed in same directory where we have html page created and image file is named as smiley.gif):

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta name="keywords" content="HTML, headTag, title, link, script">
<title>HTML Head Demo</title>
<link rel="stylesheet" type="text/css" href="mystyle.css">
<script type="text/javascript">
</script>
<style type="text/css">
body {background-color:white;}
p {color:black;}
</style>
</head>
<body>

<img src="smiley.gif" alt="Smiley face" width="42" height="42">

</body>
</html>
```

HTML Division

The tag defines a division or a section in an HTML document. The tag is used to group block-elements to format them with CSS. The attributes are
class and style.

Note: The element is very often used together with CSS, to layout a web page.

A Simple Example to show the usage:

HTMLElementsDemo.html HTMLHeadDemo.html

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta name="keywords" content="HTML, headTag, title, link, script">
<title>HTML Head Demo</title>
<link rel="stylesheet" type="text/css" href="mystyle.css">
<script type="text/javascript">
</script>
<style type="text/css">
body {background-color:white;}
p {color:black;}
</style>
</head>
<body>
<div>
<img src="smiley.gif" alt="Smiley face" width="42" height="42">
</div>
<div>
<p>Hello!! Have a Great Day</p>
</div>
</body>
</html>
```

HTML LIST

HTML provides the means for producing three types of lists: unordered (ie., unnumbered) and ordered (ie., numbered) lists, Definition (I.e define the terms like in glossary).The UL, OL and DL elements are block elements.

Unordered Lists:

An unordered list typically is a bulleted list of items. This is probably the most common type of list on the Web. The <UL> tag opens an unordered list while </UL> closes it. Between these tags are placed list items with an <LI> tag as follows:

type : Determines the form of bullet which precedes the list item placed within an unordered list. Values are disc,square,circle

*ListDemo.html

```
<UL type="circle">
<LI>red</LI>
<LI>yellow</LI>
<LI>blue</LI>
</UL>
```

*ListDemo.html

```
<UL type="circle">
<LI>red</LI>
<LI>yellow</LI>
<LI>blue</LI>
</UL>
```

OUTPUT

- red
- yellow
- blue

Ordered Lists:

An ordered list is formatted exactly the same as an unordered list, except that <OL> tags are used instead of <UL>. In an ordered list, sequential numbers are generated automatically, as shown below:

type attribute Determines the form of bullet which precedes the list item placed within an unordered list. Values are 1,a, A,i,I.

*ListDemo.html

```
<OL type="1">
<LI>red</LI>
<LI>yellow</LI>
<LI>blue</LI>
</OL>
```

OUTPUT

1. red
2. yellow
3. blue

Definition

This contains a series of terms and definitions, and would typically be used in a glossary.

*ListDemo.html

```
<dl>
  <dt>Sump</dt>
  <dd>A place where water completely fills
  the cave passage</dd>
  <dt>Rift</dt>
  <dd>A vertical fracture in the rock, created
   by geological stress</dd>
  <dd>A passage formed along such a fracture,
  usually tall and narrow</dd>
  <dt>Abseil</dt>
  <dt>Rappel</dt>
  <dd>To descend a rope using a device to control
  speed</dd>
</dl>
```

OUTPUT:

Sump
A place where water completely fills the cave passage
Rift
A vertical fracture in the rock, created by geological stress
A passage formed along such a fracture, usually tall and narrow
Abseil
Rappel
To descend a rope using a device to control speed

Ask a doubt *(Misuse of 'Ask a Doubt' Section will be dealt as per the Terms & Conditions of Campus Commune)*

Open Doubts | **Closed Doubts**

DIVJYOT MAKKAR

In the first example my paragraph and the next line are enclosed in the<a> tag,then why is the outcome not underlined?

about 12 days ago

DIVYA PANICKER

It is because **href** attribute is not used. When **href** is used the text enclosed in the <a> tag becomes an underlined link that can be clicked to navigate to a new page. The url of this new page is the value of the href attribute written within " ".

9.3. HTML Table

A table is created using the <table> tag and a closing </table> tag. A table consists of rows and columns.

Attributes of the <table> tag

•border - takes a numeric value denoting the thickness of the border around the table.

•cellspacing - takes a numeric value denoting how much space to put between cells.

•cellpadding - takes a numeric value denoting how much padding to put between what is in the cells and the cell walls.

For example

*TableDemo.html

```
<html>
<head>
<title>HTML Table Demo</title>
</head>
<body>
<table border="1" cellspacing="2" cellpadding="2">
<!-- table content goes here -->
</table>
</body>
</html>
```

Table rows

Table rows are created using the <tr> tag which should go inside the <table> tag. Use an opening <tr> tag and its closing tag </tr> for every row in the table.

for example

*TableDemo.html

```
<table border="1" cellspacing="2" cellpadding="2">
<tr></tr>
<tr></tr>
</table>
```

Table cells

Table cells are created using the <td> tag which should go inside the <tr> tag. Use an opening <td> tag and its closing tag </td> for every cell in the table.

*TableDemo.html

```
<table border="1" cellspacing="2" cellpadding="2">
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</table>
```

Adding content to the table

Now that we have a structure for our table, we can add some content to it. The content of a table is placed inside the table's cells..so inside the <td> tags in the table.

*TableDemo.html

```
<table border="1" cellspacing="2" cellpadding="2">
<tr>
<td>First row, first column</td>
<td>First row, second column</td>
</tr>
<tr>
<td>HTML</td>
<td>Javascript</td>
</tr>
<tr>
<td>CSS</td>
<td>PHP</td>
</tr>
</table>
```

Output

First row, first column	First row, second column
HTML	Javascript
CSS	PHP

Also you can add headings & captions to tables, male cells span multiple rows & columns, align the content within cells, and more.

Empty cells

Sometimes you may not need to place content inside of a table cell, instead you just need the cell to be there. But a table cell may not appear unless it has content inside of it.

You can include empty cells in a table by placing the   character entity inside a table cell.

*TableDemo.html

```
<table border="1" cellspacing="2" cellpadding="2">
<tr>
<td>Four cells in this table</td>
<td>One is empty</td>
</tr>
<tr>
<td>Three are filled</td>
<td> </td>
</tr>
</table>
```

Output:

Four cells in this table	One is empty
Three are filled	

Headings

You can display headings in a table using the <th> tag. A heading is an emphasized cell in a table. Use the <th> tag just like the <td> tag.

*TableDemo.html

```
<table border="1" cellspacing="2" cellpadding="2">
<tr>
<th>Web development</th>
<th>Software development</th>
</tr>
<tr>
<td>HTML</td>
<td>C++</td>
</tr>
<tr>
<td>PHP</td>
<td>Fortran</td>
</tr>
</table>
```

Output:

Web development	Software development
HTML	C++
PHP	Fortran

Captions

A caption is added to a table using the <caption> tag which should be placed right below the <table> tag before the first <tr> tag.

TableDemo.html

```
<table border="1" cellspacing="2" cellpadding="2">
<caption>Computer languages</caption>
<tr>
<th>Web development</th>
<th>Software development</th>
</tr>
<tr>
<td>HTML</td>
<td>C++</td>
</tr>
<tr>
<td>PHP</td>
<td>Fortran</td>
</tr>
</table>
```

Output

Computer languages

Web development	Software development
HTML	C++
PHP	Fortran

Aligning table content

You can align table content using the align attribute inside the <td> tag. This attribute can take the values

left - to align content to the left

center - to align content to the center

right - to align content to the right.

Just as you can set the alignment of table content, you can set the alignment of the table itself. This is done the same way as setting the alignment of table content.

Spanning multiple/rows columns

Table column spanning

Why should all rows have the same number of columns in a table? There may be situations where you need a cell or cells to span more than one column. This is achieved using the colspan attribute of the <td> tag. The colspan attribute takes a numeric value indicating how many columns a cell should span.

*TableDemo.html

```
<table border="2" cellspacing="4" cellpadding="10"
width="30%" align="left">
<caption>Web Languages</caption>
<tr>
<th width="40%" align="center">Client-side</th>
<td align="center">HTML</td>
<td align="center">CSS</td>
<td align="center">Javascript</td>
</tr>
<tr>
<th align="center">Server-side</th>
<td align="center">ColdFusion</td>
<td align="center" colspan="2">PHP</td>
</tr>
</table>
```

Output

Web Languages

Client-side	HTML	CSS	Javascript
Server-side	ColdFusion	PHP	

Table row spanning

You can make a cell span more than one row using the rowspan attribute of the <td> tag. The rowspan attribute takes a numeric value indicating how many rows it should span.

*TableDemo.html

```
<table border="2" cellspacing="4" cellpadding="10"
width="30%" align="left">
<caption>Web Languages</caption>
<tr>
<th align="center" rowspan="2">Web languages</th>
<td align="center">HTML</td>
<td align="center">CSS</td>
</tr>
<tr>
<td align="center">Javascript</td>
<td align="center">PHP</td>
</tr>
<tr>
<tr>
<th align="center" rowspan="2">Software languages</th>
<td align="center">C</td>
<td align="center">C++</td>
</tr>
<tr>
<td align="center">FORTRAN</td>
<td align="center">Visual Basic</td>
</tr>
<tr>
</table>
```

Output

Web Languages

Web languages	HTML	CSS
	Javascript	PHP
Software languages	C	C++
	FORTRAN	Visual Basic

Ask a doubt *(Misuse of 'Ask a Doubt' Section will be dealt as per the Terms & Conditions of Campus Commune)*

Open Doubts | **Closed Doubts**

DIVJYOT MAKKAR

In the example of spanning a row <tr> tag has been started twice which is also underlined red.does it have a significance?

about 12 days ago

VSKCHAITANYA CHALAMCHERLA

By giving <tr> , you can observe in the output that **gap between the two successive rows is increased.**. for more clarity just try by giving 3 to 5 "<tr> " and verify output..

9.4. HTML Forms

HTML Forms are required when you want to collect some data from the user and process it.

For example registration information: Name, address, date of birth, contact number, mail-id etc.

The <form> tag is used to create an HTML form:

```
<body>
<form>
..
input elements
..
</form>
</body>
```

A form will take input from the user using different form controls and form elements and then the data is passed to server for processing.

- Users interact with forms through "form controls".
- Each control has both an initial value and a current value.
- A control's initial value may be specified with the control element's "value" attribute.
- The control's current value is first set to the initial value. Thereafter, the control's current value may be modified through user interaction and

scripts.

- A control's initial value does not change. Thus, when a form is reset, each control's current value is reset to its initial value.

HTML defines the following control types:

Buttons

HTML provides three types of buttons:

- Submit buttons: When activated, a submit button submits a form.A form may contain more than one submit button.
- Example for submit button:

```
<form name="input" action="demo_form_action.asp" method="get">
Username: <input type="text" name="user">
<input type="submit" value="Submit">
</form>
```

- <input type=”submit” value=”submit”> is used to create submit buttons.
- reset buttons: When activated, a reset button resets all controls to their initial values.
- <input type=”reset” value=”reset”> is used to create submit buttons.
- push buttons: Push buttons have no default behavior. Each push button may have scripts associated with the element's events attributes. When an event occurs (e.g., the user presses the button, releases it, etc.), the associated script is executed. e.g:<button name=”edit” value=”edit”> is used to create buttons.

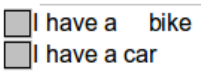
Checkboxes

- Checkboxes are on/off switches that may be toggled by the user.
- The defaultvalue of checkbox control is off, but when user selects a checkbox, the control becomes on.
- ”checked” attribute can be used when we want a default value to be selected.
- Checkboxes allow users to select several values from the set. The input element is used to create a checkbox control.

Example:

How the HTML code above looks in a browser:

```
<form name="input" action="demo_form_action.asp" method="get">
<input type="vehicle" value="Bike">I have a Bike<br>
<input type="vehicle" value="Car">I have a Bike
</form>
```



Radio buttons :

- Radio buttons are also on/off switches that may be toggled by the user.
- They are similar to checkboxes, but only one option can be selected from the set.
- e.g : <input type=”radio” name=”color” value=”red”><input type=”radio” name=”color” value=”blue”> can be used to create a radio button control with two values as red and blue.

Example:

```
<form>
<input type="radio" name= "sex" value="male">I have a Bike<br>
<input type="radio" name ="sex" value="Female">I have a Bike
</form>
```

Menus

- Menus are used when we want to choose a option from set of available options.
- The select element is used to create a menu, and option element is used to define options of that menu.
- e.g: <select> <option> red </option> <option> blue </option> </select> can be used to create a menu with two options as red and blue.

Text Fields & Textareas

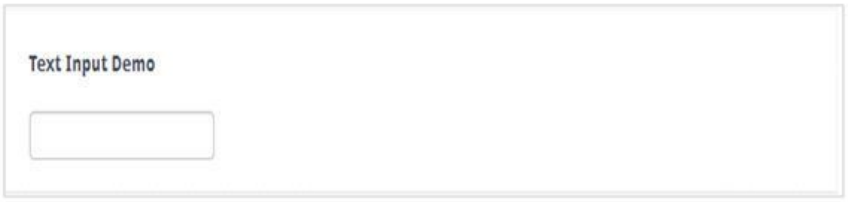
When it comes to text controls within forms there are a few different elements available to obtain data. Text fields and textareas are specifically used to gather text or string based data. This may include data containing passages of text content, passwords, telephone numbers, and so forth.

Text Fields

One of the primary elements used to obtain text from users is the input element. The input element uses the type attribute to determine what type of information is to be captured within the specific control. The most popular type attribute value is text, which denotes a single line text input. Along with setting a type attribute it is also best practice to give an input a name attribute as well. The name attribute is used as the name of the

control and is submitted along with the input's data to the server.

```
<input type ="text" name="sample_text_field">
```



Textarea

Another element used to capture text based data is the textarea element. The textarea element differs from the text input in that it is for larger passages of text spanning multiple columns. The textarea also has start and end tags which can wrap plain text. Since the textarea element only accepts one type of value the type attribute doesn't apply here, however the name attribute is still in effect.

```
<textarea name="sample_textarea">Sample textarea</textarea>
```

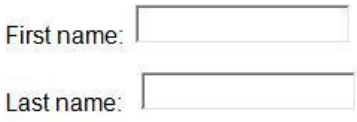
Text input

- HTML provides two types of controls that allow users to input text.
- The input element creates a single-line input control and textarea element creates a multi-line input control.

Example:

```
<form>
First name: <input type="text" name="firstname"><br>
Last name: <input type="text" name="lastname">
</form>
```

The above HTML code looks in a browser as below:



File select

This control type allows the user to select files so that their contents may be submitted with a form.
The *input* element is used to create a file select control.

Drop Down Lists:

Drop down lists are a perfect way to provide users with a long list of options in a usable manner. Outputting every state within the country on a page with radio buttons would create a rather cluttered and daunting list. Drop down menus provide the perfect venue for a long list of options.

To create a drop down menu the select and option elements are used. The select element will wrap all of the different menu options marked up using the option element. Then you can apply the name attribute to the select element.

Each individual option within the menu needs to be coded using the option element. The option element will wrap the text to be included within the menu. Additionally, the option element will include the value attribute, specific to each individual option.
As with the checked Boolean attribute with radio buttons and checkboxes, drop down menus can use the selected Boolean attribute to preselect an option for users.

```
<select name="day">
  <option value="Friday" selected>Friday</option>
  <option value="Saturday">Saturday</option>
  <option value="Sunday">Sunday</option>
</select>
```

Multiple Selections

Using the standard drop down list and adding the Boolean attribute multiple to the select element allows a user to choose more than one option from the menu. Additionally, using the selected Boolean attribute on more than one option element within the menu will preselect multiple options. The height and width of the select element can be controlled using CSS and should be adjusted appropriately to allow for multiple selections. It may also be worth mentioning to users that to choose multiple options they will need to hold down the shift key while clicking to make their selections.

```
<select name="day" multiple>
  <option value="Friday" selected>Friday</option>
  <option value="Saturday">Saturday</option>
  <option value="Sunday">Sunday</option>
</select>
```

Form Buttons

After a user inputs the requested information, buttons allow them to put that information into action. Most commonly, a submit button is used to process the data. A reset button is often used to clear data.

Submit Button

Users click the submit button to process data after filling out a form. The submit button uses the input element with a type attribute of either submit or image. The submit attribute value is the most common as it is simple to use and provides the most control. The image attribute value is used specifically to set an image as a submit button, however with the use of CSS the need to use an image has greatly diminished.
To determine the verbiage to be used within the button, the value attribute is used. Using CSS properties such as background, border-radius, box-shadow, and others, the button can then be styled to any specific desire.

```
<input type="submit" name="submit" value="Submit Form">
```

Reset Button

To take the complete opposite action from submitting a form, users may also reset a form using the reset button. The reset button code works just like that of the submit button, however it uses the reset value for the type attribute.
Reset buttons are becoming less and less popular because they offer a very strong action, often undesired by both websites and users. Users may spend quite a bit of time filling out a form only to click the reset button accidentally thinking it was the submit button. Chances of a user filling out the form again thereafter are small. Before using a reset button think of any potential consequences.

```
<input type="reset" name="reset" value="Reset Form">
```

Other Inputs

Outside all other previously stated choices the input element does have a few other use cases. Two of which include passing hidden data and attaching filings during form processing.

Hidden Input

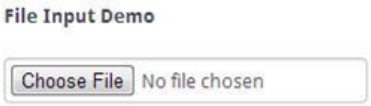
Hidden inputs provide a way to pass data to the server without displaying it to users. Hidden inputs are typically used for tracking codes, keys, or other information not pertinent to the users but helpful to the website overall. This information is not displayed on the page, however it could be seen by viewing the source code of a page. That said, it should not be used for sensitive or secure information.
To create a hidden input you use the hidden value for the type attribute. Additionally, you pass along the appropriate name and value.

```
<input type="hidden" name="tracking_code" value="abc_123">
```

File Input

To allow users to add a file to a form, much like that of attaching a file to an email, the file value for the type attribute is used. The file input is most commonly seen to upload pictures or files to social networks or applications.
Unfortunately, stylizing the file input is a tough task with CSS. Each browser has its own default rendering of how the input should look and doesn't provide much control to override the default styling. JavaScript and other solutions can be built to allow for file input, but they are slightly more difficult to construct.

```
<input type="file" name="file">
```



Organizing Form Elements

Knowing how to capture data with inputs is half the battle. Organizing form elements and controls into a usable manner is the other half. Forms are not worth much unless users understand what is asked of them and how to provide the requested information. Using labels, fieldsets, and legends we can better organize forms and guide users to completing the end task.

Label

Labels provide captions, or headings, for form elements. Created using the label element, labels should include descriptive text, of which describes the input or control it pertains to. Labels should include a for attribute. The value of the for attribute should be the same as the value of the id attribute included within the form element the label corresponds to. Matching up the for and id values ties the two elements together, allowing users to click on the label to get focused within the proper form element.

```
<label for="username">Username</label>
<input type="text" name="username" id="username">
```

When using labels with radio buttons or checkboxes the input element can be wrapped within the label element. Doing so allows omission of the for and id attributes.

```
<label><input type="radio" name="day" value="Friday" checked> Friday</label>
<label><input type="radio" name="day" value="Saturday"> Saturday</label>
<label><input type="radio" name="day" value="Sunday"> Sunday</label>
```

Fieldset

Fieldsets group form controls and labels into organized sections. Much like a div or other structural element, the fieldset is a block level element that wraps related elements, however specifically within a form for better organization. Fieldsets by default also include a border outline that can be modified using CSS.

```
<fieldset>
  <label for="username">Username</label>
  <input type="text" name="username" id="username">
  <label for="password">Password</label>
  <input type="text" name="password" id="password">
</fieldset>
```

Ask a doubt *(Misuse of 'Ask a Doubt' Section will be dealt as per the Terms & Conditions of Campus Commune)*

Open Doubts | **Closed Doubts**

ABDUL HAQ

Can any please tell the purpose of **<label>**

about 1 month ago

TEJASWINI

The <label> tag defines a label for an <input> element.

The <label> element does not render as anything special for the user. However, it provides a usability improvement for mouse users, because if the user clicks on the text within the <label> element, it toggles the control.

ALURI REDDY

The HTML <label> tag is used to adda label to a form control like text, textarea etc.
it provides a usability improvement for mouse users, because if the user clicks on the text within the <label> element, it toggles the control.

9.5. CSS

Introduction

Css stands for cascading style sheets. Styles define how html tags are displayed or formatted in the browser. Styles can be applied to html tags to format the look and feel of the html pages. By defining CSS based styling, the styling/formatting code can be separated from html code.

Styles are normally defined in an external file called style sheets which have a .css extension. The stylesheets are then imported into the html file. CSS works by defining rules how a selector (html tag) can be displayed.

CSS Syntax

A CSS rule set consists of a selector and a declaration block. The general syntax of a CSS rule set looks like this:

Selector {property:value; property:value; . . .}
Selector – points to the html element that needs to be formatted
Property – points to the specific html element's attribute that is being formatted
Value – is the value given to the attribute that is used for formatting

More than one attribute/property of the html element can be specified in the CSS ruleset. The different sets of properties/value pairs are separated

by a semi-colon. To improve the readability the property:value pairs can be written on separate lines.

```
p{color:blue; text-align:left;}
```

The ruleset above formats the color of the text to blue inside a html <p> tag and aligns the text to left.

```
p
{
color:red;
font-style:italic;
font-family:”arial”;
}
```

The ruleset above formatted the color of text to red inside a html <p> tag, and formats the text to italics and sets to Arial font.

Grouping Selectors

In style sheets there are often elements with same styling. To minimize code the selectors can be grouped. To group selectors separate each selector with a comma.

Objective: Group selectors to apply same styling to multiple html elements

```
h2, h3, p
{
 color:grey;
 font-weight:italics;
 font-family:verdana;
}
```

html

```
<html>
    <head>
        <Title>Content Alignment and CSS</Title>
        <link href='style.css' type='text/css' rel='stylesheet' />
    </head>
    <body>
        <div class='container'>
        <div class='left'>
            <h2>Few Indian Beverages</h2>
            <ul>
                <li id='first'>Lassi</li>
                <li>Chhaas</li>
                <li>Aamras</li>
                <li>Sugarcase juice</li>
            </ul>

        </div>
        <div class='right'>
            <h3>Popular Indian Beverages</h3>
            <ul>
                <li style='color: green'>Coffee</li>
                <li>Tea</li>
                <li>Jal-jeera</li>
                <li>Sharbat</li>
            </ul>
        </div>
        </div>
        <p>Above examples are taken for demonstrating the usage of style sheets and
        its implementation.</p>
    </body>
</html>
```

More About CSS Selectors

The selectors can be used to find all or specific html elements and apply styling to a single html element or to more than one or the entire html elements at once.

Id selector

The Id selector uses the "id" attribute of an html tag to find the specific html element. The id should be a unique in an html page. Using the unique id then you can find a single html element and apply styling.

In order to use the id selector, first the html element is given a unique id. The id can be a character, string or a combination of characters and numbers. The id should not start with a number. Then to find the element using the id write hash character followed by the id of the element.

```
#p1,h1
{
text-align: left;
color:red;
}
```

The ruleset above will find the html para and heading elements with the specified id values and apply the styling.

Class selector

The class selector uses the html "class" attribute of an html tag to find the html elements. Unlike "id" selector "class" is generally not given a unique value. Thus the "class" attribute can be used to apply styling to more than one html element that have the same value for the class attribute.

In order to use the class selector, first the html element is given a class name. The name of the class can be a character, string or a combination of characters and numbers. The class name cannot start with a number. Then to find the emelemt using the class name write a period character followed by the class name.

```
p.center, tr.center
{
text-align: center;
color:red;
}
```

The ruleset above finds all the html para and table row elements with class="center" and applies the styling.

A class selector can be created free of a tag name in which case all the tags that have the same class name will be styled similarly.

```
.right
{
text-align:right;
color:blue;
font-family:"calibri";
}
```

The id selector can be applied to one , unique element compared to the class selector which can be applied to several different elements.

Objective: *To apply style using id and class attributes.*
'.' in front of each tag defines a class and corresponding class is mentioned in the element used in HTML page.

We can use '#' to define style for a id and give the corresponding id for the element in HTML page. Id for each element in HTML page is always unique. No two elements in an HTML file should have the same id.

```
.left {
  width: 39%;
  float:left;
  text-align:left;
  display:inline;
}
.right {
   width: 39%;
   float:right;
   text-align:left;
   display:inline;
}
.container {
   width: 80%;
   margin:auto;
}
#first {
color: red;
}
```

```
<html>
    <head>
        <Title>Content Alignment and CSS</Title>
        <link href='style.css' type='text/css' rel='stylesheet' />
    </head>
    <body>
        <div class='container'>
        <div class='left'>
            <h3>Few Indian Beverages</h3>
            <ul>
                <li id='first'>Lassi</li>
                <li>Chhaas</li>
                <li>Aamras</li>
                <li>Sugarcase juice</li>
            </ul>

        </div>
        <div class='right'>
            <h3>Popular Indian Beverages</h3>
            <ul>
                <li>Coffee</li>
                <li>Tea</li>
                <li>Jal-jeera</li>
                <li>Sharbat</li>
            </ul>
        </div>
        </div>
    </body>
</html>
```

Output:

Element selector

The element selector selects html elements based on the element name. When styling is applied based on the element name all elements that have the same name will be applied with the same styling.

```
p
{
text-align: center;
color: red;
}
```

For example the ruleset above will format all the <p> elements in the html document with the styling mentioned in the ruleset.

Objective: *Use of tags to apply style in stylesheets*

The styling is applied by using the html element name. In this case the style is applied using the style is applied to the html element h3.

```
.left {
  width: 39%;
  float:left;
  text-align:left;
  display:inline;
}
.right {
   width: 39%;
   float:right;
   text-align:left;
   display:inline;
}
.container {
   width: 80%;
   margin:auto;
}
#first {
color: red;
}
h3 {
color: blue;
}
```

```
<html>
    <head>
          <Title>Content Alignment and CSS</Title>
          <link href='style.css' type='text/css' rel='stylesheet' />
    </head>
    <body>
          <div class='container'>
          <div class='left'>
                <h3>Few Indian Beverages</h3>
                <ul>
                      <li id='first'>Lassi</li>
                      <li>Chhaas</li>
                      <li>Aamras</li>
                      <li>Sugarcase juice</li>
                </ul>

          </div>
          <div class='right'>
                <h3>Popular Indian Beverages</h3>
                <ul>
                      <li>Coffee</li>
                      <li>Tea</li>
                      <li>Jal-jeera</li>
                      <li>Sharbat</li>
                </ul>
          </div>
          </div>
    </body>
</html>
```

Output

Ask a doubt *(Misuse of 'Ask a Doubt' Section will be dealt as per the Terms & Conditions of Campus Commune)*

Open Doubts | **Closed Doubts**

SATYANARAYAN K

Where do we need to write the CSS code?In the same notepad file as the html file or Any Special Software Required??

about 14 days ago

SUMAN SINGH

you can write it in the html code itself using this code in <head> section

<style type="text/css">

your css code

</style>

You can also write the css in another file, save it with .css extension and import this file in your html file

SUMAN SINGH

you can write it in the html code itself using this code in <head> section

<style type="text/css">

your css code

</style>

You can also write the css in another file, save it with .css extension and import this file in your html file

9.6. Inserting CSS into Html

There are 3 ways of inserting style sheets:

- External style sheet
- Internal style sheet
- Inline style

External style sheet

External style sheet is ideal when applying style to many pages. The styling code is maintained in a single file externally. The look of an entire website can be changed with changes done to just this single file.

An html page can link to an external style sheet using the <link> tag. The <link> tag is defined under the <head> tag.

```
<head>
<link rel="stylesheet" type="text/css" href="mystyle.css">
</head>
```

The import rule can also be used to import an external css file.

```
<head>
<style type="text/css">
   @import url(mystyle.css);
</style>
</head>
```

The external style sheet can be written in any text editor. The file should contain only the styling code and should not contain any html tags. The style sheet must be saved with a .css extension.

Objective: *To apply style using an external CSS file and then include it in the HTML page.*

Create a CSS file named style.css (or any other name of your choice) and write the code shown in the image below.

'.' in front of each tag defines a class and corresponding class is mentioned in the element used in HTML page.

style.css

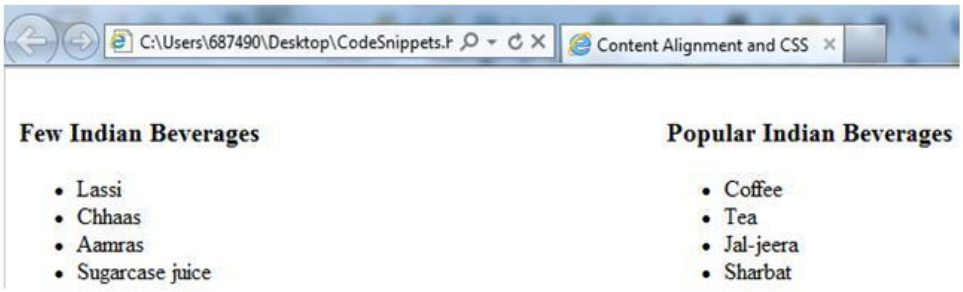
```
.left {
  width: 39%;
  float:left;
  text-align:left;
  display:inline;
}

.right {
   width: 39%;
   float:right;
   text-align:left;
   display:inline;
}

.container {
   width: 80%;
   margin:auto;
}
```

Create a HTML file in the same folder where the CSS file is created. HTML file should contain the following code.

The style sheet is imported in the head section using the html link tag.

```
<html>
   <head>
         <Title>Content Alignment and CSS</Title>
         <link href='style.css' type='text/css' rel='stylesheet' />
   </head>
   <body>
         <div class='container'>
         <div class='left'>
               <h3>Few Indian Beverages</h3>
               <ul>
                     <li>Lassi</li>
                     <li>Chhaas</li>
                     <li>Aamras</li>
                     <li>Sugarcase juice</li>
               </ul>

         </div>
         <div class='right'>
               <h3>Popular Indian Beverages</h3>
               <ul>
                     <li>Coffee</li>
                     <li>Tea</li>
                     <li>Jal-jeera</li>
                     <li>Sharbat</li>
               </ul>
         </div>
         </div>
   </body>
</html>
```

Output:



Internal style sheet

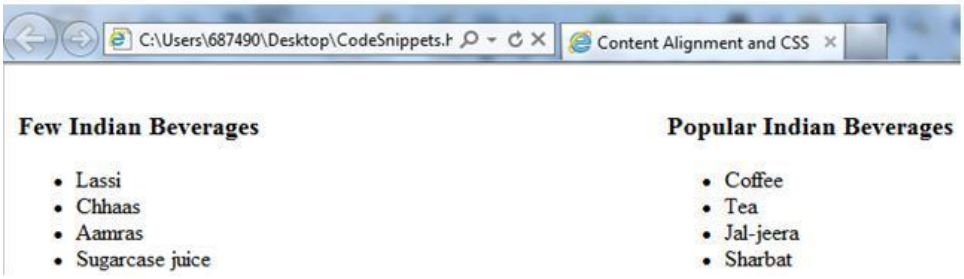
Internal style sheets are used when applying unique styles to single documents. Internal styles are written inside <head> tags using <style> tag.

```
<head>
<style>
.left
{
color:blue;
text-align:left;
font-family:"calibri";
}
</style>
</head>
```

Objective: *To apply internal style to div tags, so that two div tags are displayed side by side. The style applied is present inside style tags in the head section of HTML file.*

```
<html>
    <head>
        <Title>Content Alignment and CSS</Title>

        <style type='text/css'>
            .left {
              width: 39%;
              float:left;
              text-align:left;
              display:inline;
            }

            .right {
               width: 39%;
               float:right;
               text-align:left;
               display:inline;
            }

            .container {
               width: 80%;
               margin:auto;
            }

        </style>
    </head>
    <body>
        <div class='container'></div>
        <div class='left'>
            <h3>Few Indian Beverages</h3>
            <ul>
                <li>Lassi</li>
                <li>Chhaas</li>
                <li>Aamras</li>
                <li>Sugarcase juice</li>
            </ul>

        </div>
        <div class='right'>
            <h3>Popular Indian Beverages</h3>
            <ul>
                <li>Coffee</li>
                <li>Tea</li>
                <li>Jal-jeera</li>
                <li>Sharbat</li>
            </ul>
        </div>
        </div>
    </body>
</html>
```

Output:



Inline styles

Inline styles are written as part of the html tag itself. The style attribute of the html tag is used and can contain any css property. This way of styling should be used only sparingly and when the other two ways of inserting syles is not feasible since it is least flexible.

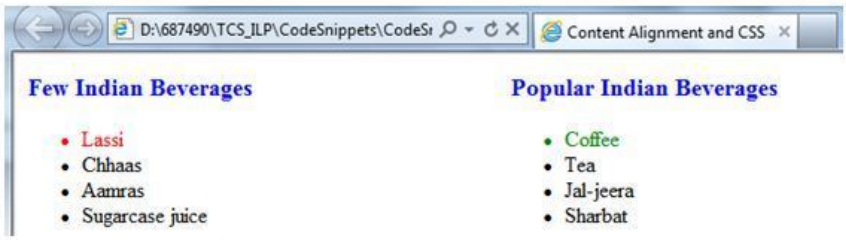
```
<p style="color:blue; text-alight:left"> This is a paragraph </p>
```

Objective: *To apply styles at the html element/tag level or inline styling.*

Html:

```
<html>
    <head>
        <Title>Content Alignment and CSS</Title>
        <link href='style.css' type='text/css' rel='stylesheet' />
    </head>
    <body>
        <div class='container'>
        <div class='left'>
            <h3>Few Indian Beverages</h3>
            <ul>
                <li id='first'>Lassi</li>
                <li>Chhaas</li>
                <li>Aamras</li>
                <li>Sugarcase juice</li>
            </ul>

        </div>
        <div class='right'>
            <h3>Popular Indian Beverages</h3>
            <ul>
                <li style='color: green'>Coffee</li>
                <li>Tea</li>
                <li>Jal-jeera</li>
                <li>Sharbat</li>
            </ul>
        </div>
        </div>
    </body>
</html>
```

Output:



Overriding the Styles

Multiple styles, if applied to same element will reflect in following order:

1. Style attribute applied at element/tag level will have top most preference.
2. Style applied in the head section using style tag of HTML page will be preferred next.
3. External style will have the last preference.

What if I forcefully want to give preference to a style defined for an element in the external style sheet? Code shown below helps us to achieve it.

Objective: *Overriding the styles applied*

```
<html>
    <head>
        <Title>Content Alignment and CSS</Title>
        <link href='style.css' type='text/css' rel='stylesheet' />
    </head>
    <body>
        <div class='container'>
        <div class='left'>
            <h2>Few Indian Beverages</h2>
            <ul>
                <li id='first'>Lassi</li>
                <li>Chhaas</li>
                <li>Aamras</li>
                <li>Sugarcase juice</li>
            </ul>

        </div>
        <div class='right'>
            <h3>Popular Indian Beverages</h3>
            <ul>
                <li style='color: green'>Coffee</li>
                <li>Tea</li>
                <li>Jal-jeera</li>
                <li>Sharbat</li>
            </ul>
        </div>
        </div>
        <p>Above examples are taken for demonstrating the usage of style sheets and
        its implementation.</p>
    </body>
</html>
```

```
.left {
  width: 39%;
  float:left;
  text-align:left;
  display:inline;
}

.right {
   width: 39%;
   float:right;
   text-align:left;
   display:inline;
}

.container {
   width: 80%;
   margin:auto;
}

#first {

color: red;

}

h3 {

color: blue !important;

}

h2, h3, p {

color: grey;
font-weight: italics;
font-family: verdana;

}
```

Output:

Ask a doubt *(Misuse of 'Ask a Doubt' Section will be dealt as per the Terms & Conditions of Campus Commune)*

9.7. Different Tools To Create UI

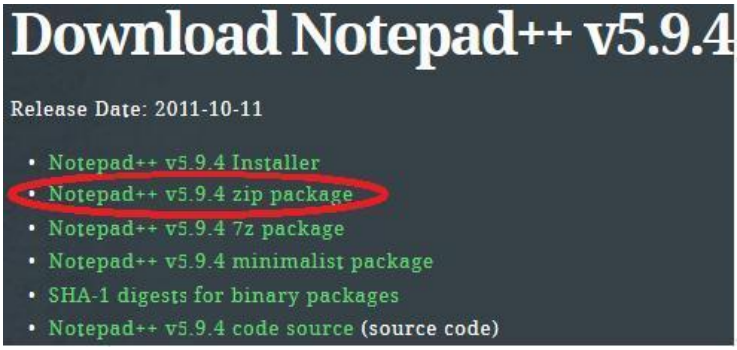
Notepad++

Notepad++ is a free source code editor and Notepad replacement that supports several programming languages.

Download Notepad++:

To download notepad++, open web browser and enter the following URL http://notepad-plus-plus.org/download/v5.9.4.html

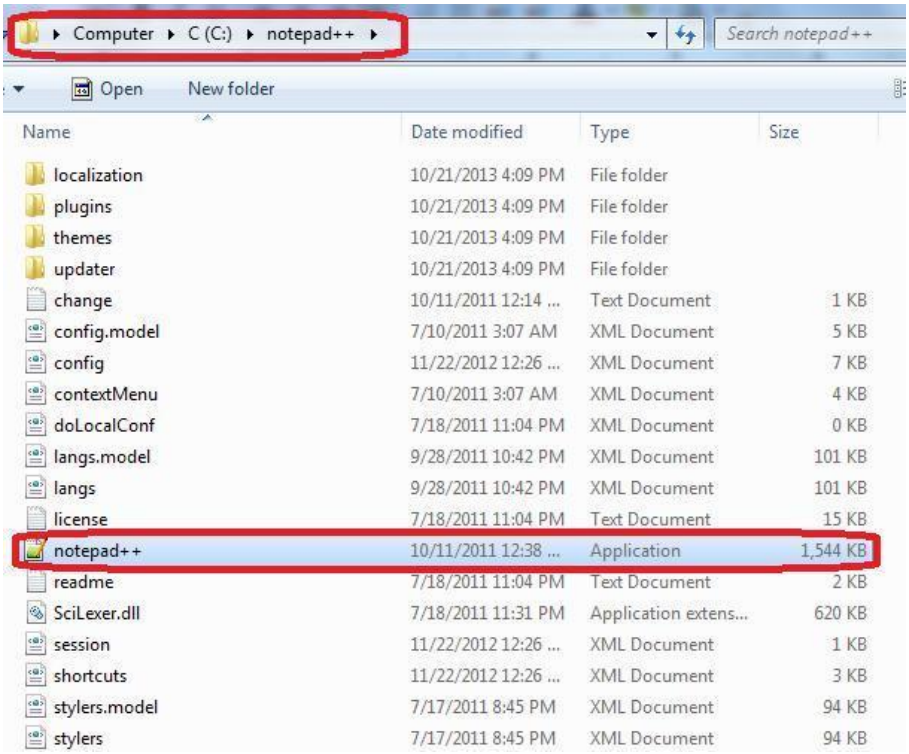
Wait till the entire page gets loaded. Though the latest version of Notepad++ is available we recommend you to use version 5.9.4. Click Notepad++ v5.9.4 zip package.



You may download and place the archive (zip) at any location of the hard disk of your system. Then the archive needs to be extracted using any of the archiving tools provided by your OS. The installation of notepad++ is done.

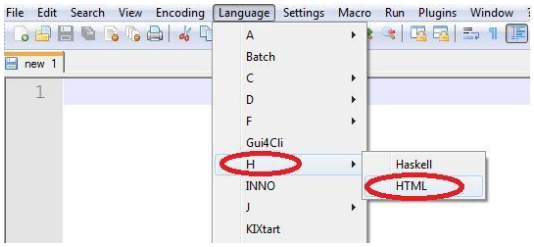
1. Open Notepad++

Open notepad++ from the Notepad++ installation directory. The Notepad++ installation directory will have a notepad++ application as highlighted below.



Demo Of HTML in Notepad++

Course 1 - We shall now write a sample HTML code. Select HTML from the Language Tab

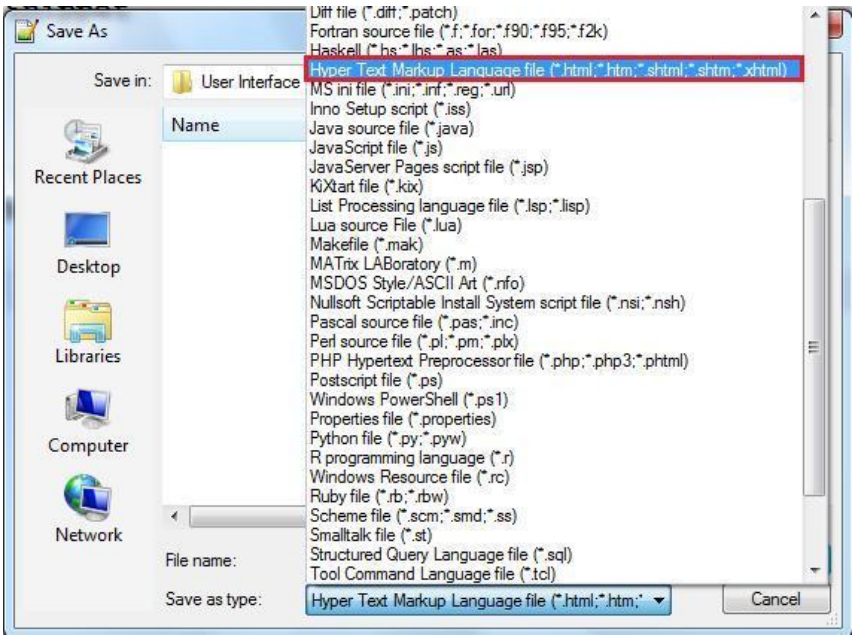


Course 2 - Write HTML code in the editor area.

```
<!-- To display content on the web page and
        change the title of window-->

<HTML>
 <HEAD>
  <TITLE>
    <!-- This name appears in window's title bar. -->
    My First HTML Code Snippet
  </TITLE>
 </HEAD>
  <BODY>
   <!-- This line show up on the web page. -->
    This is my first HTML page.
  </BODY>
</HTML>
```

If you observe the code, all the HTML tags are colored in blue and comments in green. This is an advantage with Notepad++. So we can clearly make out in case of any typing mistakes.

Course 3 - Save this file as .html/.htm type and file name can be anything of your choice. Let us name it MyFirstHTML

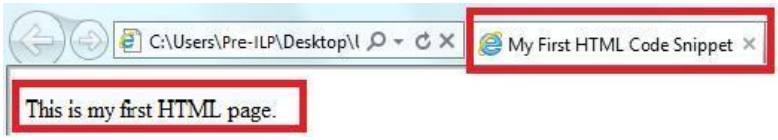


To execute the saved HTML file, you can follow two ways:

1. Open from saved location

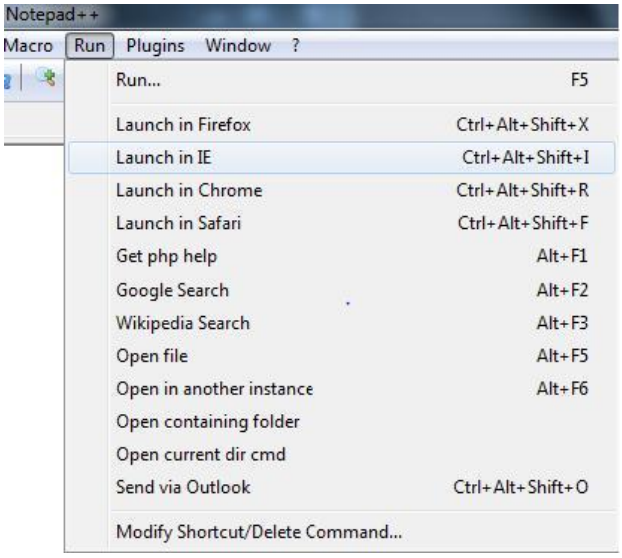
Course 1 - Now your HTML file is saved and if you open the location where it is saved, the icon of the file would be your default browser icon as following:

Course 2 - Double click the file. The file is opened with your default browser.



2. Launch from Notepad++

a) Select Run from the Menu Bar and click on Launch in IE/Chrome/Firefox/Safari. Note that you need to save the file before executing it.



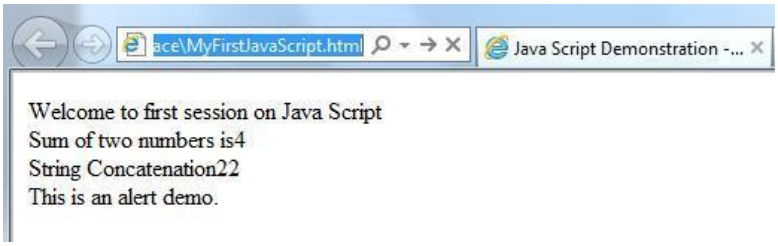
Demo of JavaScript in Notepad++

We shall now see the demo of JavaScript

```
<!-- To be able to display data on page using JavaScript.
    Also perform mathematical and string operations using JS-->
<HTML>
 <BODY>
 <!-- The following title appears in window's title bar. -->
  <TITLE>Java Script Demonstration</TITLE>
  <script language='javascript'>
    document.write("Welcome to first session on Java Script");
    </script>
    <br/>
     <script language='javascript'>
    document.write("Sum of two numbers is"+(2+2));
    </script>
    <br/>
     <script language='javascript'>
    document.write("String Concatenation"+ 2+2);
    </script>
    <br/>
 </BODY>
</HTML>
```

Here, you can see that attributes of script tag are distinctly colored, if you try to define any other attribute which is not predefined, you will find it in black.

Save this file as .html/.htm type and execute in the same way as we did for HTML.



Demo of CSS in Notepad++

```
<!-- To apply style to div tags,so that teo div tags are displayed side by side.
The style applied is present as a style tag in the head section of HTML file.-->
<HTML>
 <HEAD>
  <TITLE>Content Alignment and CSS</TITLE>
  <STYLE type='text/css'>
   .left{
        width: 39%;
        float: left;
        text-align:left;
        display:inline;
        }
   .right{
       width: 39%;
       float: right;
       text-align: left;
       display: inline;
       }
   .container{
        width: 80%;
        margin:auto;
        }
   </STYLE>
 </HEAD>
 <BODY>
   <div class='container'>
   <div class='left'>
        <h3>Few Indian Beverages</h3>
        <ul>
            <li>Lassi</li>
            <li>Chhaas</li>
            <li>Aamras</li>
            <li>Sugarcane Juice</li>
        </ul>
   </div>
   <div class='right'>
        <h3>Popular Indian Beverages</h3>
        <ul>
            <li>Coffee</li>
            <li>Tea</li>
            <li>Jal-Jeera</li>
            <li>Sharbat</li>
        </ul>
   </div>
   </div>
  </BODY>
</HTML>
```

Save the program and execute it in the same way as we did for HTML and JavaScript.

Refer the below link which would help you debug your web pages opened with Internet Explorer:

http://msdn.microsoft.com/library/gg589507(VS.85).aspx

Ask a doubt *(Misuse of 'Ask a Doubt' Section will be dealt as per the Terms & Conditions of Campus Commune)*

9.8. Practice Problems

General Instructions for these exercises:

1. There are total five pages to be created.
2. First one is home page with links to below three pages.
3. From below pages, on successful submit, navigate to welcome page.
4. Welcome page should have link to home page.
5. Place CSS and Javascript code in separate files.
6. Use effective styling as per your choice.
7. Use div tags and table for better layout.
8. Maintain consistency for all pages.

- **Add a doctor**

Design a UI page / screen to capture the following details when registering a new doctor in HMS.

First name, Middle name Last name, Registration No, Date of Birth , Gender(use radio button), Email, Specialization , Qualification, contact number, Address, Visiting Time, Visiting Days(multiple selection should be allowed), NoOfPatientsPerDay, DepartmentNo.

Indicate the mandatory fields on the page. All fields except middle name and address are mandatory.

Use below mentioned sample values for drop down/check boxes:

Visiting Days – Mon,Tues,Wed,Thurs, Fri,Sat,Sun.

Visiting Time – Morning, Afternoon, Evening

- **Add a department**

Design a UI page / screen to capture the following details when adding a new department in HMS.

Department name, Department Description(use text area), No. Of Doctors, Number of general rooms, Number of ac rooms, Number of non ac rooms , Number of ICU , Cost of general room, Cost of ac room, Cost of non ac room, Cost of each ICU.

All fields are mandatory.

- **Add a patient**

Design a UI page / screen to capture the following details when adding a new patient in HMS.

First Name, Last Name , Date of birth, Gender(Use radio button), Father/Spouse name, blood group(dropdown), Contact number, Visited earlier(check box)

All fields are mandatory except contact number.

Indicate mandatory fields on the page / screen.

Use below mentioned values in the dropdown :-

Blood Group – O+,O-,AB+,AB-,A+,A-,B+,B-,Others

Click here to download the solutions

Ask a doubt *(Misuse of 'Ask a Doubt' Section will be dealt as per the Terms & Conditions of Campus Commune)*

9.9. Reference Links & Videos

Reference Links

To know more on HTML and CSS, visit the URLs:

http://en.wikipedia.org/wiki/HTML

http://www.howtocreate.co.uk/tutorials/html/basics

http://www.htmlgoodies.com/tutorials/html_401/

http://www.w3schools.com/html/default.asp

http://html.net/tutorials/html/

http://www.tutorialspoint.com/html/what_is_new_in_html4.htm

http://www.w3resource.com/html/doctype-HTML.php

http://www.alternetwebdesign.com/htmltutorial/lesson3.htm

http://www.w3schools.com/css/DEFAULT.asp

http://learnlayout.com/

Related Videos

Ask a doubt *(Misuse of 'Ask a Doubt' Section will be dealt as per the Terms & Conditions of Campus Commune)*

Open Doubts | **Closed Doubts**

SHIVSHANKAR MURALI

While taking up the test, I got an error message stating that the webpage could not start and followed by some Raw process output text. Unable to take up the test. What should I do now ?

I have a screenshot of the error. Unable to upload :(

about 17 days ago

DIVYA PANICKER

Contact ilp or campus commune team.

Java

Course Completion Quiz

10.Designing Web Pages Part 2 (JavaScript and JQuery)

Objective

- This material will take you through concepts of JavaScript. After successful completion of this course the associate should be able to use basic JavaScript functionality.
- jQuery overview and installation
- Calling jQuery library functions
- jQuery Selectors, CSS and Attributes
- jQuery DOM Traversing
- jQuery Events

10.1. Introduction

JavaScript is a scripting language that is used in most of the modern HTML pages.
It gives a dynamic touch to them.

JavaScript is used to change the content of HTML elements dynamically. It can also be used to read the content of HTML elements and validate them and eventually show corresponding message to the user.

Just like we have tags for HTML elements, JavaScript is written inside <script></script> tags. By convention script tags are placed inside head tag, although it is not necessary to do that.

Below is a sample of JavaScript code, in which an alert method is used which is used to display messages as pop-up.

```
<html>
    <head>
        <script>
            alert('Hello');
        </script>
    </head>
    <body>
    </body>
</html>
```

OUTPUT:

When the browser loads the web page, the scripts are also loaded. Thus in the example above, as soon as the web page is loaded the browser ran the script and the alert method was executed.

The JavaScript can be included into the web page in two ways.
We can either write the code in script tag or write the code in another external file, save it with .js extension and include in the web page.

For example we can write "alert('hello');" in a text file and save it as myfile.js
Then we can include the js file in the web page.

<script src="myfile.js"></script>

Variables

Variables are used to store data and manipulate it, just like we had in Java.
A variable in JavaScript is declared as:
var variableName=data;

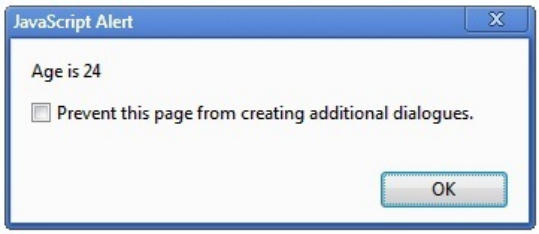
To declare a variable, **"var"** keyword is used. The data type of the variables is determined by the type of data they hold.

eg:

var x="George"; //string
x='a'; //char
x=5; //int
x=4.5 //float
x=true; //boolean
x=new Array(2,4,6,3); //array of integer

```
<html>
    <head>
        <script>
            var x="George";
            alert('Name is '+x);
            x=24;
            alert('Age is '+x);
        </script>
    </head>
    <body>
    </body>
</html>
```

OUTPUT:



Here we can see that same variable was able to store different values at different times. So java script is type independent.

Events

An event is defined as an action performed over an HTML element.

On various actions performed by mouse or keyboard JavaScript code can be called.

We will discuss about four basic events. **onclick, onsubmit, onchange, onselect**

onclick event occurs when we click on any HTML element. lets see an example of clicking a button.

```
<html>
    <head>
    </head>
    <body>
        <button onclick="alert('You clicked on me')">Click Me</button>
    </body>
</html>
```

OUTPUT:

When we click on a button, **onclick** event occurrs and the alert method is called as per the code.

onsubmit event occurs when the form is being submitted. Here is a sample code.

```
<html>
    <head>
    </head>
    <body>
        <form action="" method="post" onsubmit="alert('You are about to submit the form')">
        <input type="text">
        <input type="submit">
        </form>
    </body>
</html>
```

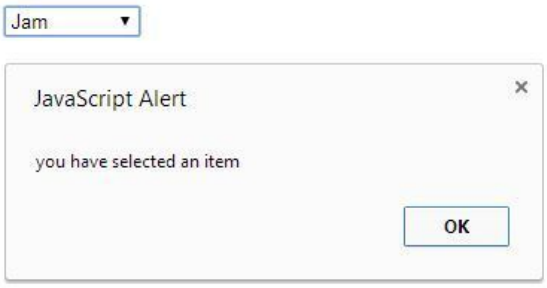
OUTPUT:

When submit button is pressed an **onsubmit** event occurs.

onchange event occurs whenever there is a change in the content of the particular element. For example:

```
<!DOCTYPE html>
<html>
<head>
</head>
<body>
<select  onchange="alert('you have selected an item')">
<option selected="selected">Select Item</option>
<option>Jam</option>
<option>Sauce</option>
<option>cream</option>
</select>
</body>
</html>
```

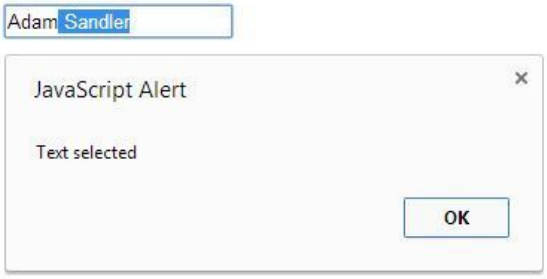
OUTPUT:



onselect event occurs when text of an input tag is selected. Sample code:

```
<html>
<head>
</head>
<body>
<input type="text" onselect="alert('Text selected')">
</body>
</html>
```

OUTPUT:



When the text in the text box is selected **onselect** event occurs and according to the code the alert method was called.

Functions

All java script codes written inside script tag gets executed after the page is loaded by the browser. If we want to execute specific functionality on various events we should put the JavaScript codes inside functions and call those function at the occurrence of events.

You must be familiar with method calling in Java. It is similar to that.
To write a function , the key word "function" is used, followed by the function name and the arguments.

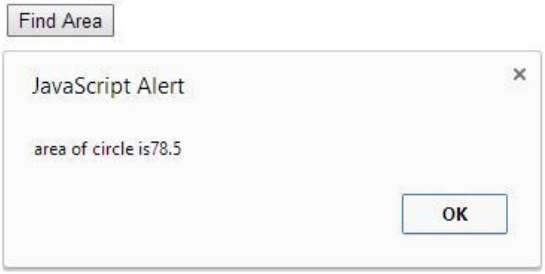
```
funciton myFunction(argument1, argument2....){

  var value=10;
  return value;
}
```

A variable declared inside a function becomes a local variable and thus can be used within the function only. A variable declared outside a function becomes a global variable and thus can be accessed by any function.

Sample Code:

```
<html>
<head>
    <script>
        var pi=3.14;

        function calculateArea(radius){
            alert('area of circle is'+(pi*radius*radius));
        }
    </script>
</head>
<body>
    <button onclick="calculateArea(5)">Find Area</button>
</body>
</html>
```

OUTPUT:

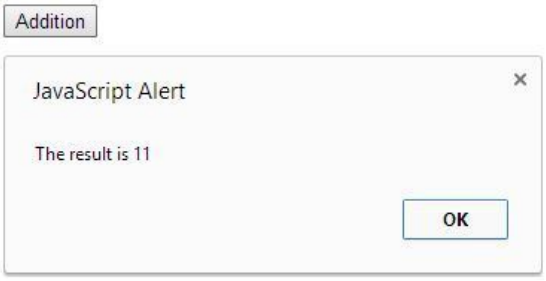


In the above example, a logic for calculating area of a circle is placed. Then on the event of clicking on the button, the function was called and a number 5 was passed as an argument. The area of the circle was shown through an alert box.

A function can call another function and return value to the calling function.
Here is an example.

```
<html>
<head>
    <script>
        function calculate(x,y,z){
            var r=0;
            if(z==1)
            r=add(x,y);
            alert('The result is '+r);
        }

        function add(x,y){
            return x+y;
        }
    </script>
</head>
<body>
    <button onclick="calculate(5,6,1)">Addition</button>
</body>
</html>
```

OUTPUT



In this example add function was called by the calculate function and it returned the sum the numbers.

Ask a doubt *(Misuse of 'Ask a Doubt' Section will be dealt as per the Terms & Conditions of Campus Commune)*

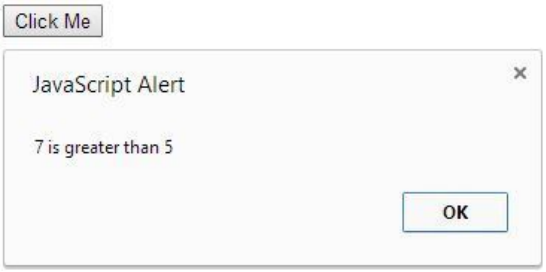
10.2. Conditional Operations

To compare between two expressions we need to do conditional operations.JavaScript support conditional operations like if-else, if-esle if and nested ifelse

Below is an example of if-else if :

```
<html>
    <head>
        <script>
            function compare(x,y){
            if(x>y){
                alert(x+' is greater than '+y);
                }
            else if(x<y){
                alert(y+' is greater than '+x);
                    }
            else{
                alert('both the numbers are equal');
                }
                    }
        </script>
    </head>
    <body>
        <button type="button" onclick="compare(5,7)">Click Me</button>
    </body>
</html>
```

OUTPUT



Here two numbers are compared and it is determined which number is greater.

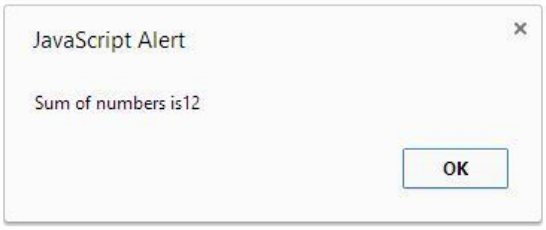
For Loop

In JavaScript there are many ways to iterate, one of them is to use a for loop.It is used in the same way as in Java.

Sample Code:

```
<html>
<head>
    <script>
    var x=new Array(2,4,6);
    var sum=0;
    for(var i=0;i<x.length;i++){
        sum=sum+x[i];
    }
    alert('Sum of numbers is'+sum);
    </script>
</head>
<body>
</body>
</html>
```

OUTPUT



Here we iterated through the array of integers and calculated the sum of those numbers.The alert method showed the result on the screen.

This can also be achieved by placing the code inside a function and then calling the function through an event on HTML element.

PopUp Boxes

In JavaScript we can create small pop-up windows to either show message or get input from user. JavaScript provides three type of popup boxes:

Alert Box, Confirm Box and Prompt Box.

You have already seen alert box functionality, it is used to show messages as pop up.

Confirm box, as the name suggests, is used for confirmations. It returns either true or false when we click on Ok and Cancel respectively.

Syntax: Event="return confirm(Message)";

when we write **return** the boolean value returned will determine whether the form will be submitted or not.

Sample Code:

```
<html>
<head>
</head>
<body>
    <form action="/myservlet.do">
        <input type="text">
        <input type="submit" onclick="return confirm('Are you sure you want to submit?')">
    </form>
</body>
</html>
```

OUTPUT

When you click on "Cancel", false will be returned and the form will not be submitted.
When you click on "OK", true will be returned and the form will proceed with the submission.

Prompt Box is used when we want an input from the user through the popup box.

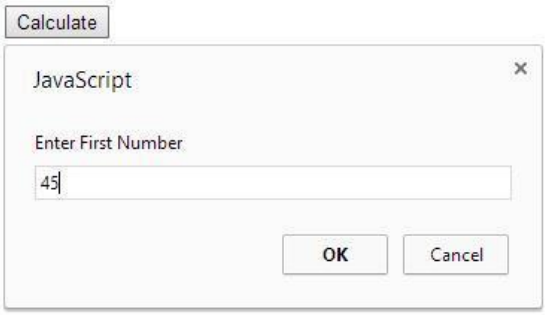
Syntax: prompt(Message,default value);

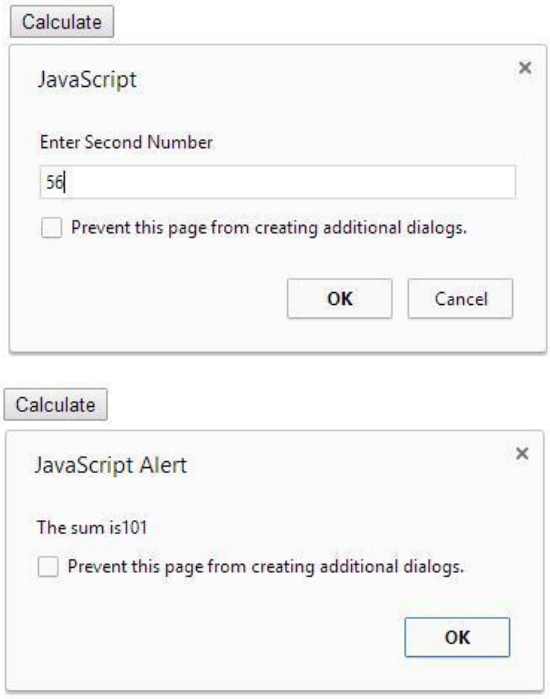
The default value is not necessary to give and can be skipped. The return type is string that we enter in the box.

Sample code:

```
<html>
<head>
</head>
<script >
    function add(){
        var x=prompt("Enter First Number");
        var y=prompt("Enter Second Number");
        alert('The sum is'+(parseInt(x)+parseInt(y)));
    }

</script>
<body>
<button onclick="add()">Calculate</button>
</body>
</html>
```

OUTPUT





In the code we used inbuilt function parseInt() to convert string into int.

When Cancel is pressed on a prompt box, it returns null.

Ask a doubt *(Misuse of 'Ask a Doubt' Section will be dealt as per the Terms & Conditions of Campus Commune)*

Open Doubts | **Closed Doubts**

SASIVINEETHA GATTUPALLE

In the sample code of the prompt box the variables are x and y. Then wat are parseInt(x) and parseInt(y)? Do they specify anything? Thank You.

about 17 days ago

MADDI

The value that is accepted in the prompt box is a **String**. In order to change it to the data type needed, we use the wrapper classes. In the code given above, we need to find sum of integers. So we have used parseInt().to convert x and y to **int.**

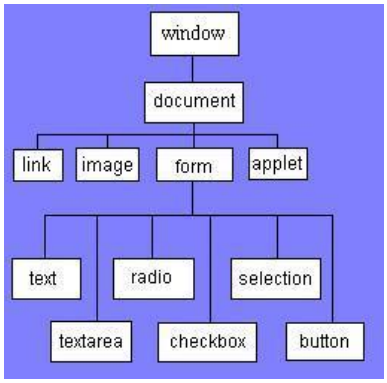
SWARNADEEP SAHA

Hi.. Maddi,
I think there is no integer datatype in JavaScript. All arithmetic data are shared "number" variable which is a 64 bit floating point. (IEEE-754).

10.3. Document Object Model and Validations

Document Object Model

When a browser loads a web page, it creates the Document Object Model of the page.
Its a tree of HTML elements and we use it to traverse to various elements of the HTML page.



In the JavaScript function we can access the elements of the page either by traversing through the elements or by using some of the methods of these objects.

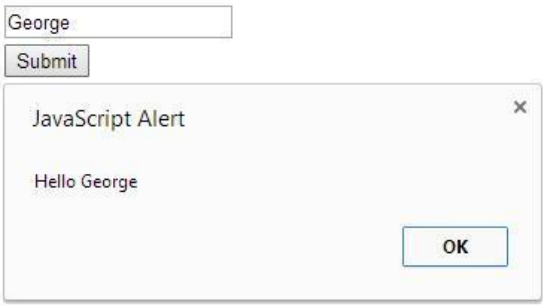
getElementById:

This method is used to access an element using the unique id we assign to it while creating the page.

Sample Code:

```
<html>
<head>
</head>
<script >
    function sayHello(){
        var x=document.getElementById("name");
        alert('Hello '+x.value);
    }

</script>
<body>
<form name="myForm" action="/register.do">
    <input name="firstName" id="name">
    <br>
    <input type="submit" onclick="sayHello()">
</form>
</body>
</html>
```

OUTPUT



In the above example var x=document.getElementById("name") was used to access the text field having id=name.
Here the **document** object's method **getElementById** was used.
The same could be achieved by writing

var x=document.forms["myForm"]["name"].value where we use the 2D form of document model.

or

var x=document.myForm.firstName.value where we cascade through elements using their names.

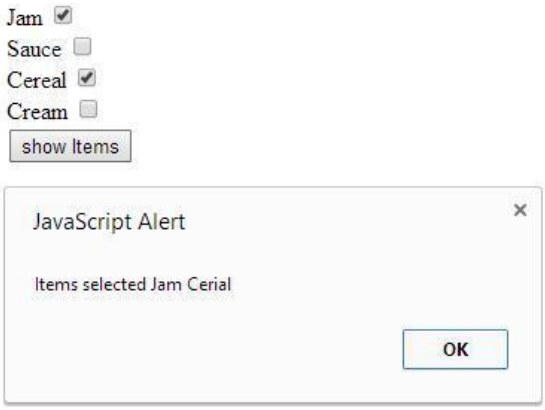
getElementsByName:

This method is used to get all the elements with the given name.

Sample code:

```
<html>
<head>
    <script>
    function printItems(){

        var x=document.getElementsByName("name");
        var items="";
        for(var i=0;i<x.length;i++){
            if(x[i].checked)
                items=items+x[i].value+" ";
        }
        alert('Items selected '+ items);
    }
    </script>
</head>
<body>
<form action="" name="myForm">
    Jam <input type="checkbox" name="name" value="Jam"><br>
    Sauce <input type="checkbox" name="name" value="Sauce"><br>
    Cereal <input type="checkbox" name="name" value="Cerial"><br>
    Cream <input type="checkbox" name="name" value="Cream"><br>
    <button onclick="printItems()">show Items</button>
</form>
</body>
</html>
```

OUTPUT



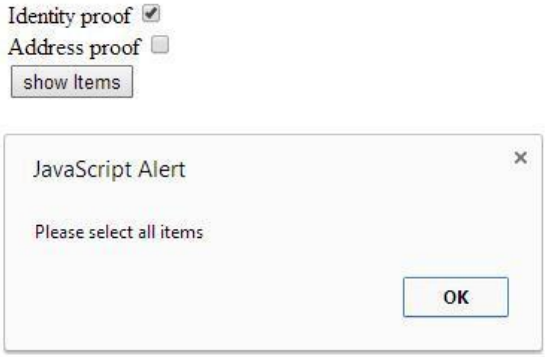
Here all the elements by that name are stored as array of elements in variable x. After that we can we can iterate through x and get all the elements and their value.

getElementsByTagName:

This method is used to get all the elements with the name of the tag.
Sample code:

```
<html>
<head>
    <script>
    function printItems(){

        var x=document.getElementsByTagName("input");
        var bool=true;
        for(var i=0;i<x.length;i++){
            if(!x[i].checked){
                bool=false;
                break;}
        }
            if(!bool)
        alert('Please select all items');
    }
    </script>
</head>
<body>
    Identity proof <input type="checkbox" name="name" value="Identity proof"><br>
    Address proof <input type="checkbox" name="name" value="Address proof"><br>
    <button onclick="printItems()">Proceed</button>
</body>
</html>
```

OUTPUT



Here all the elements by the Tag name "input" are stored as array of elements in variable x. After that we can we can iterate through x and get all the elements and check whether they are checked or not.

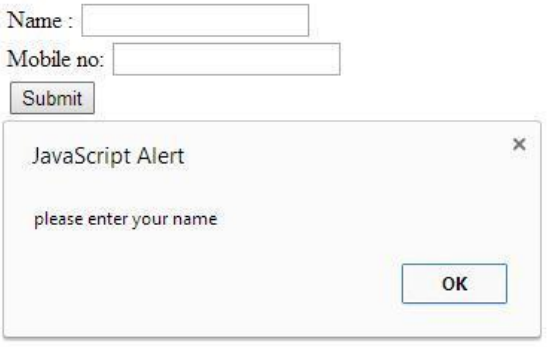
Validations

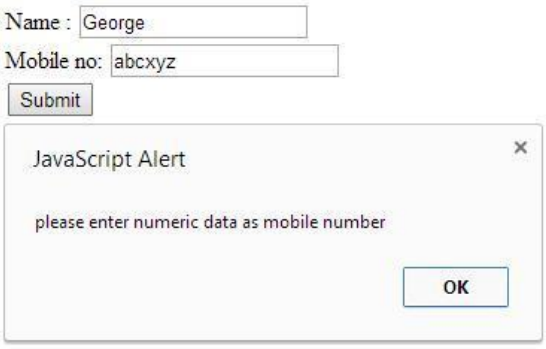
We can use JavaScript to validate the data entered by the users. For instance, we can check whether user has entered into the text field or not, whether user has enter a string in a number field, whether user has exceeded the world limit etc.

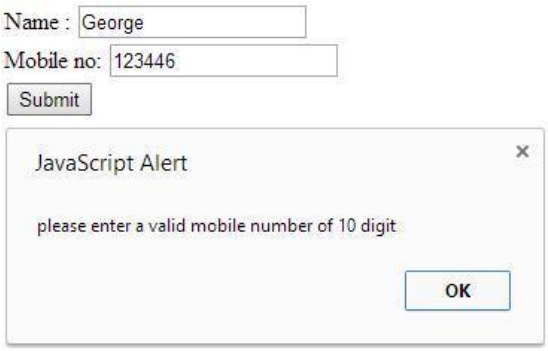
In the example below we will be validating a form based on the criteria mentioned above.

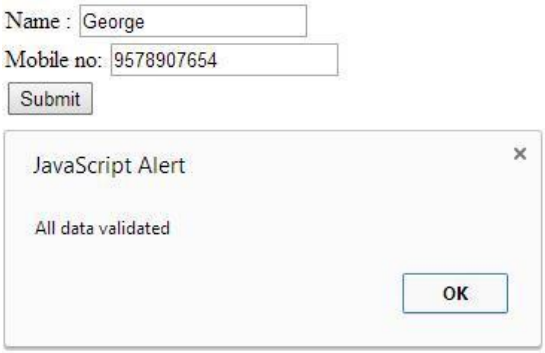
```
<html>
<head>
</head>
<script >
    function validate(){
        var x=document.getElementById("name").value;
        var y=document.getElementById("number").value;

        if(x==""){
            alert('please enter your name');
            return false;
        }
        if(y==""){
            alert('please enter your number');
            return false;
        }
        if(isNaN(y)){
            alert('please enter numeric data as mobile number');
            return false;
        }
        if(y.length!=10){
            alert('please enter a valid mobile number of 10 digit');
            return false;
        }
        alert('All data validated')
       return true;
    }

</script>
<body>
<form action="/register.do" onsubmit="return validate()">
    Name     : <input type="text" id="name"><br>
    Mobile no: <input type="text" id="number"><br>
    <input type="submit" value="Submit">
</form>
</body>
</html>
```

OUTPUT









Here first validations were did to find out whether the fields were left blank or not.
If any field was left blank, proper message would be shown.
Then the mobile number was validated for being a number first. When it is confirmed that its a number only, then its length was checked. If the length of the number is other than 10 then that means the number was not valid, so the error message was shown in the alert box.

Radio button validation

To check whether a radio button or check box is checked or not, simply first get the element by name, id or tag name and write the following conditional code:

if(x.checked), where x is the radio element or check box element. Thus we can validate each HTML element using JavaScript.

Sample code:

```
<html>
<head>
</head>
<script >
    function checkRadio(){
        var x=document.getElementsByName("gender")
        for(var i=0;i<x.length;i++){
            if(x[i].checked)
                return true;
        }
        alert('please select gender');
       return false;
    }

</script>
<body>
<form action="/register.do" >
    Male: <input type="radio" name="gender" value="male">
    Female: <input type="radio" name="gender" value="female"><br>
    <input type="submit" onclick="return checkRadio()">
</form>
</body>
</html>
```

OUTPUT

The above sample code that checks whether a radio button is checked or not.

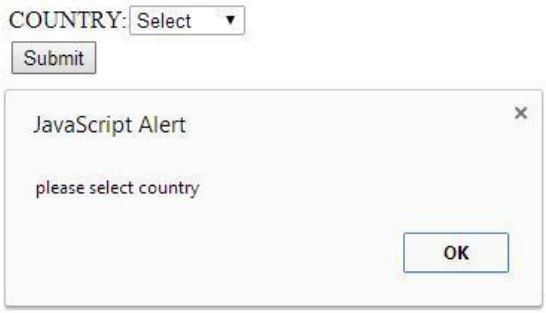
As soon as JavaScript finds that the radio button is selected it return true and the form proceeds with submission otherwise false is returned and the form is not submitted. A corresponding message is also shown through the alert box.

Select box validation

Now let us see how to validate a select box.

```
<html>
<head>
</head>
<script >
    function checkRadio(){
        var x=document.myForm.country;
        if(x.selectedIndex==0)
            {
                alert('please select country');
                return false;
            }
        alert(x.value);
       return true;
    }

</script>
<body>
<form name="myForm" action="/register.do" >
    COUNTRY:<select name="country">
    <option value="">Select</option>
    <option value="France">France</option>
    <option value="Germany">Germany</option>
    <option value="India">India</option>
    <option value="Italy">Italy</option>
    <option value="Kenya">Kenya</option>

    </select><br>
    <input type="submit" onclick="return checkRadio()">
</form>
</body>
</html>
```

OUTPUT



In the above example first the select element was fetched through the document object. Then the index of the selected option was checked. The index starts from 0. So after clicking the submit button if the index of selected option remains same, then that would mean the user did not select any option.

If the selected index is not 0 then the code would show the value of selected option through alert box. i.e "x.value" here would give the selected option value.

Ask a doubt *(Misuse of 'Ask a Doubt' Section will be dealt as per the Terms & Conditions of Campus Commune)*

10.4. jQuery Overview

jQuery is a fast and concise JavaScript Library created by John Resig in 2006 with a nice motto: Write less, do more. jQuery simplifies HTML document traversing, event handling, animating, and Ajax interactions for rapid web development.

jQuery is a JavaScript tool kit designed to simplify various tasks by writing less code. Here is the list of important core features supported by jQuery:

DOM manipulation: jQuery made it easy to select DOM elements, traverse them and modifying their content by using cross-browser open source selector engine called Sizzle.

Event handling: jQuery offers an elegant way to capture a wide variety of events, such as a user clicking on a link, without the need to clutter the HTML code itself with event handlers.

AJAX Support: jQuery helps you a lot to develop a responsive and feature-rich site using AJAX technology.

Animations: jQuery comes with plenty of built-in animation effects which you can use in your websites.

Lightweight: jQuery is very lightweight library - about 19KB in size.

Cross Browser Support: jQuery has cross-browser support, and works well in IE 6.0+, FF 2.0+, Safari 3.0+, Chrome and Opera 9.0+

Latest Technology: jQuery supports CSS3 selectors and basic XPath syntax.

jQuery Installation

To start using jQuery on your web site, download the jQuery library from jQuery.com. There are two versions of jQuery available for downloading. Both versions can be downloaded from jQuery.com.

1) Production version - this is for your live website because it has been minified and compressed

2) Development version - this is for testing and development (uncompressed and readable code)

The jQuery library is a single JavaScript file, and you reference it with the HTML <script> tag (notice that the <script> tag should be inside the <head> section):

```
<head>
        <script src="jquery-1.10.2.min.js"></script>
</head>
```

How to call jQuery library functions?

You can include jQuery library in your HTML file as follows:

```
<html>
    <head>
        <title>The title</title>
        <script type="text/javascript" src="jquery-1.3.2.min.js"></script>
        <script type="text/javascript">
            // you can add our javascript code here
        </script>
    </head>
    <body>
        Your HTML code goes here...
    </body>
</html>
```

jQuery reads or manipulates the document object model (DOM), we need to make sure that we start adding events etc. as soon as the DOM is ready.

If you want an event to work on your page, you should call it inside the $(document).ready() function. Everything inside it will load as soon as the DOM is loaded and before the page contents are loaded.

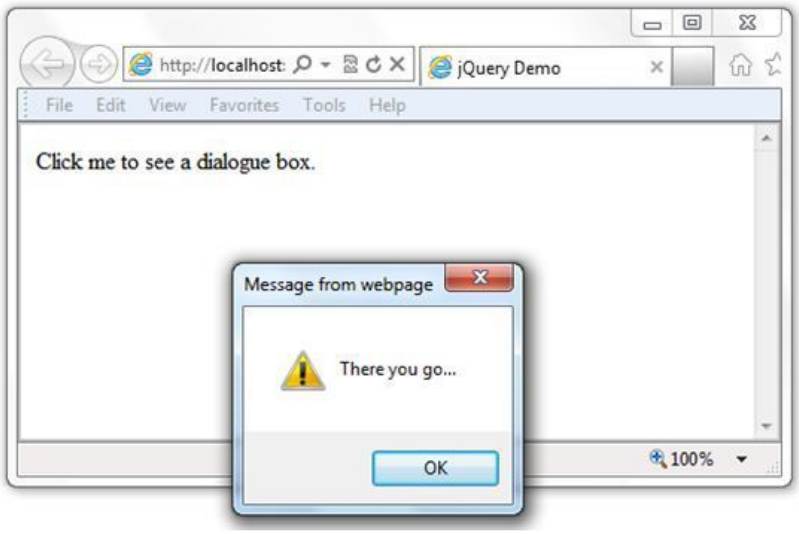
To do this, we register a ready event for the document as follows:

```
$(document).ready(function() {
    alert('Hi....');
    // write the stuff to execute when DOM is ready
});
```

To call upon any jQuery library function, use HTML script tags as shown below:

```
<html>
<head>
<title>jQuery Demo</title>
   <script type="text/javascript" src="jquery.js"></script>

   <script type="text/javascript" language="javascript">
   $(document).ready(function() {
      $("div").click(function() {
        alert("There you go...");
      });
   });
   </script>

</head>
<body>
<div id="newdiv">
Click me to see a dialogue box.
</div>
</body>
</html>
```

When you click on the text in on the page, following result is seen:



It is better to write our custom code in the custom JavaScript file : custom.js, as follows:

```
//code inside external js file custom.js
$(document).ready(function() {
      $("div").click(function() {
        alert("There you go...");
      });
   });
```

And include custom.js file in your HTML file as follows:

```
<html>
<head>
<title>jQuery Demo</title>
   <script type="text/javascript" src="jquery.js"></script>
   <script type="text/javascript" src="custom.js"></script>
</head>
<body>
<div id="newdiv">
Click me to see a dialogue box.
</div>
</body>
</html>
```

jQuery Selectors

The jQuery library harnesses the power of Cascading Style Sheets (CSS) selectors to let us quickly and easily access elements or groups of elements in the Document Object Model (DOM).

All type of selectors available in jQuery, always start with the dollar sign and parentheses: $(). A jQuery Selector is a function which makes use of expressions to find out matching elements from a DOM based on the given criteria. The factory function $() is a synonym of jQuery() function. So

in case you are using any other JavaScript library where $ sign is conflicting with some thing else then you can replace $ sign by jQuery name and you can use function jQuery() instead of $().

The factory function $() makes use of following three building blocks while selecting elements in a given document:

1) Tag Name
Represents a tag name available in the DOM. For example $('p') selects all paragraphs in the document.

2) Tag ID
Represents a tag available with the given ID in the DOM. For example $('#some-id') selects the single element in the document that has an ID of some-id.

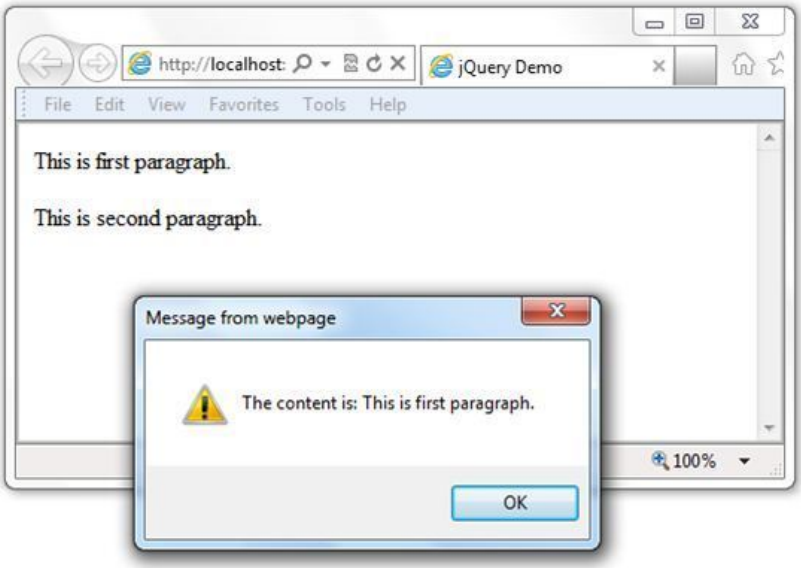
3) Tag Class
Represents a tag available with the given class in the DOM. For example $('.some-class') selects all elements in the document that have a class of some-class.

All the above items can be used either on their own or in combination with other selectors. All the jQuery selectors are based on the same principle except some tweaking.

Following is a simple example which makes use of Tag Selector. This would select all the elements with a tag name p.

```
<html>
<head>
<title>jQuery Demo</title>
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript" language="javascript">
    $(document).ready(function() {
        var pars = $("p");
        for (i = 0; i < pars.length; i++) {
            alert("The content is: " + pars[i].innerHTML);
        }
    });
</script>
</head>
<body>
<div>
    <p class="myclass">This is first paragraph.</p>
    <p id="myid">This is second paragraph.</p>
</div>
</body>
</html>
```

The result appear as show below and different alert messages are displayed for every paragraph on the page:



How to use Selectors?

The selectors are very useful and would be required at every step while using jQuery. They get the exact element that you want from your HTML document.

Following table lists down few basic selectors and explains them with examples.

Selector	Description
Name	Selects all elements which match with the given element Name.
#ID	Selects a single element which matches with the given ID
.Class	Selects all elements which match with the given Class.
Universal (*)	Selects all elements available in a DOM.
Multiple Elements E, F, G	Selects the combined results of all the specified selectors E, F or G.

Similar to above syntax and examples, following examples would give you understanding on using different type of other useful selectors:

$('*'): This selector selects all elements in the document.

$("p > *"): This selector selects all elements that are children of a paragraph element.

$("#specialID"): This selector function gets the element with id="specialID".

$(".specialClass"): This selector gets all the elements that have the class of specialClass.

$("li:not(.myclass)"): Selects all elements matched by <li> that do not have class="myclass".

$("a#specialID.specialClass"): This selector matches links with an id of specialID and a class of specialClass.

$("p a.specialClass"): This selector matches links with a class of specialClass declared within <p> elements.

$("ul li:first"): This selector gets only the first <li> element of the <ul>.

$("#container p"): Selects all elements matched by <p> that are descendants of an element that has an id of container.

$("li > ul"): Selects all elements matched by <ul> that are children of an element matched by

$("strong + em"): Selects all elements matched by that immediately follow a sibling element matched by .

$("p ~ ul"): Selects all elements matched by <ul> that follow a sibling element matched by <p>.

$("code, em, strong"): Selects all elements matched by <code> or or .

$("p strong, .myclass"): Selects all elements matched by that are descendants of an element matched by <p> as well as all elements that have a class of myclass.

$(":empty"): Selects all elements that have no children.

$("p:empty"): Selects all elements matched by <p> that have no children.

$("div[p]"): Selects all elements matched by that contain an element matched by <p>.

$("p[.myclass]"): Selects all elements matched by <p> that contain an element with a class of myclass.

$("a[@rel]"): Selects all elements matched by <a> that have a rel attribute.

$("input[@name=myname]"): Selects all elements matched by <input> that have a name value exactly equal to myname.

$("input[@name^=myname]"): Selects all elements matched by <input> that have a name value beginning with myname.

$("a[@rel$=self]"): Selects all elements matched by <p> that have a class value ending with bar

$("a[@href*=domain.com]"): Selects all elements matched by <a> that have an href value containing domain.com.

$("li:even"): Selects all elements matched by <li> that have an even index value.

$("tr:odd"): Selects all elements matched by <tr> that have an odd index value.

$("li:first"): Selects the first <li> element.

$("li:last"): Selects the last <li> element.

$("li:visible"): Selects all elements matched by <li> that are visible.

$("li:hidden"): Selects all elements matched by <li> that are hidden.

$(":radio"): Selects all radio buttons in the form.

$(":checked"): Selects all checked/selected elements in the form.

$(":input"): Selects only form elements (input, select, textarea, button).

$(":text"): Selects only text elements (input[type=text]).

$("li:eq(2)"): Selects the third <li> element

$("li:eq(4)"): Selects the fifth <li> element

$("li:lt(2)"): Selects all elements matched by <li> element before the third one; in other words, the first two <li> elements.

$("p:lt(3)"): selects all elements matched by <p> elements before the fourth one; in other words the first three <p> elements.

$("li:gt(1)"): Selects all elements matched by <li> after the second one.

$("p:gt(2)"): Selects all elements matched by <p> after the third one.

$("div/p"): Selects all elements matched by <p> that are children of an element matched by .

$("div//code"): Selects all elements matched by <code>that are descendants of an element matched by .

$("//p//a"): Selects all elements matched by <a> that are descendants of an element matched by <p>

$("li:first-child"): Selects all elements matched by <li> that are the first child of their parent.

$("li:last-child"): Selects all elements matched by <li> that are the last child of their parent.

$(":parent"): Selects all elements that are the parent of another element, including text.

$("li:contains(second)"): Selects all elements matched by <li> that contain the text second.

You can use all the above selectors with any HTML/XML element in generic way. For example if selector $("li:first") works for <li> element then $("p:first") would also work for <p> element.

Ask a doubt *(Misuse of 'Ask a Doubt' Section will be dealt as per the Terms & Conditions of Campus Commune)*

10.5. jQuery Attributes

Some of the most basic components we can manipulate when it comes to DOM elements are the properties and attributes assigned to those elements.

Most of these attributes are available through JavaScript as DOM node properties. Some of the more common properties are:

- className
- tagName
- id
- href
- title
- rel
- src

Consider the following HTML mark-up for an image element:

```
<img id="myImage" src="image.gif" alt="An image" class="someClass"
    title="This is an image" />
```

In this element's mark-up, the tag name is img, and the mark-up for id, src, alt, class, and title represents the element's attributes, each of which consists of a name and a value.

jQuery gives us the means to easily manipulate an element's attributes and gives us access to the element so that we can also change its properties.

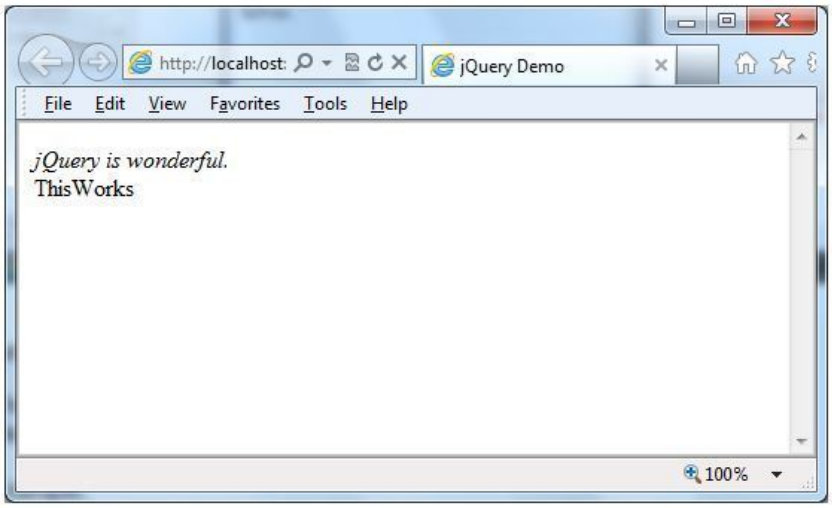
Get Attribute Value:
The attr() method can be used to either fetch the value of an attribute from the first element in the matched set or set attribute values onto all matched elements.

Example:

Following is a simple example which fetches title attribute of tag and sets it as the content of the div:

```
<html>
<head>
<title>jQuery Demo</title>
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript" language="javascript">
    $(document).ready(function() {
        var title = $("em").attr("title");
        $("#divid").text(title);
    });
</script>
</head>
<body>
    <em title="ThisWorks">jQuery is wonderful.</em>
    <div id="divid"></div>
</body>
</html>
```

You see following output when the page is displayed in the browser:



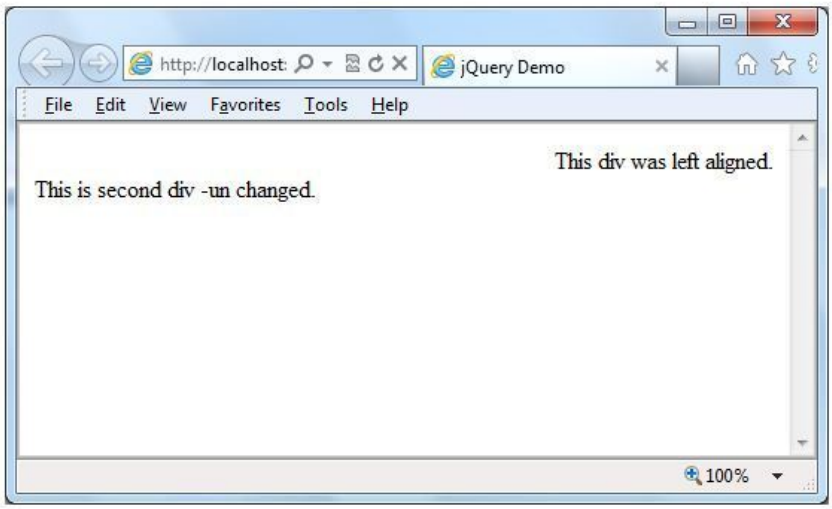
Set Attribute Value:

The attr(name, value) method can be used to set the named attribute onto all elements in the wrapped set using the passed value.

Example:
Following is a simple example which sets align attribute of first div tag:

```
<html>
<head>
<title>jQuery Demo</title>
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript" language="javascript">
    $(document).ready(function() {
        $("#div").attr("align", "right");
    });
</script>
</head>
<body>

    <div align="left" id="div">
        This div was left aligned.
    </div>
    <div id="secondDiv">
        This is second div -un changed.
    </div>
</body>
</html>
```

Output appears as below on the screen:



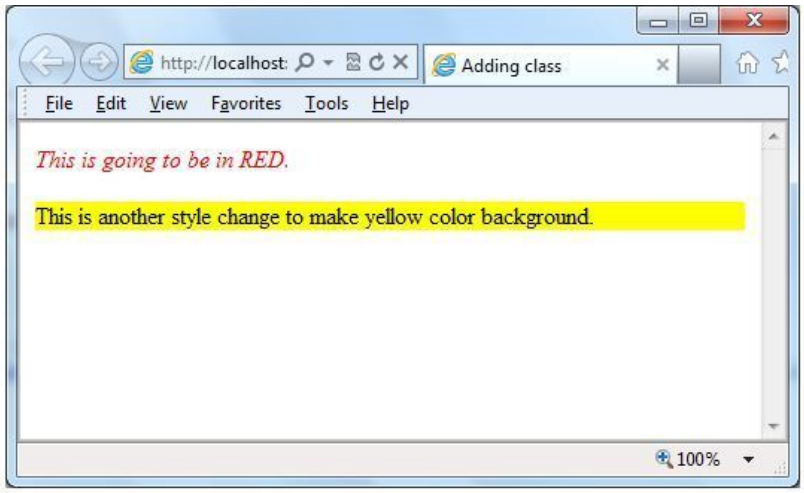
Applying Styles:

The addClass(classes) method can be used to apply defined style sheets onto all the matched elements. You can specify multiple classes separated by space.

Example:
Following is a simple example which sets class attribute of a <p> and tags:

```
<html>
<head>
<title>Adding class</title>
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript" language="javascript">
    $(document).ready(function() {
        $("em").addClass("selected");
        $("#myid").addClass("emc");
    });
</script>
<style>
.selected {
    color: red;
}

.emc {
    background: yellow;
}
</style>
</head>
<body>
<em>This is going to be in RED.</em>
<p id="myid">This is another style change to make yellow color background.</p>
</body>
</html>
```

The generated out put appears as below:



Useful Attribute Methods:

Following table lists down few useful methods which you can use to manipulate attributes and properties:

Methods	Description
attr(name)	Get the value of an attribute for the first element in the set of matched elements
attr(name, value)	Set attribute for the set of matched elements
attr(attributes)	Set attribute-value pairs.
attr(name, function(index, attr))	A function returning the value to set. this is the current element. Receives the index position of the element in the set and the old attribute value as arguments.
removeAttr(name)	Remove an attribute from each of the matched elements.
hasClass(class)	Returns true if the specified class is present on at least one of the set of matched elements.
removeClass(class)	Removes all or the specified class(es) from the set of matched elements.
toggleClass(class)	Adds the specified class if it is not present, removes the specified class if it is present.
html()	Get the html contents (innerHTML) of the first matched element.
html(val)	Set the html contents of every matched element.
text()	Get the combined text contents of all matched elements.
text(val)	Set the text contents of all matched elements.
val()	Get the input value of the first matched element.
val(val)	Set the value attribute of every matched element if it is called on <input> but if it is called on <select> with the passed <option> value then passed option would be selected, if it is called on check box or radio box then all the matching check box and radio would be checked.

Similar to above syntax and examples, following examples would give you understanding on using various attribute methods in different situation:

$("#myID").attr("custom") : This would return value of attribute custom for the first element matching with ID myID.

$("img").attr("alt", "Sample Image"): This sets the alt attribute of all the images to a new value "Sample Image".

$("input").attr({ value: "", title: "Please enter a value" }); : Sets the value of all <input> elements to the empty string, as well as sets the title to the string Please enter a value.

$("a[href^=http://]").attr("target","_blank"): Selects all links with an href attribute starting with http:// and set its target attribute to _blank

$("a").removeAttr("target") : This would remove target attribute of all the links.

$("form").submit(function() {$(":submit",this).attr("disabled", "disabled");}); : This would modify the disabled attribute to the value "disabled" while clicking Submit button.

$("p:last").hasClass("selected"): This return true if last <p> tag has associated classselected.

$("p").text(): Returns string that contains the combined text contents of all matched <p> elements.

$("p").text("Hello World"): This would set "<I>Hello World</I>" as text content of the matching <p> elements

$("p").html() : This returns the HTML content of the all matching paragraphs.

$("div").html("Hello World") : This would set the HTML content of all matching to Hello World.

$("input:checkbox:checked").val() : Get the first value from a checked checkbox

$("input:radio[name=bar]:checked").val(): Get the first value from a set of radio buttons

$("button").val("Hello") : Sets the value attribute of every matched element <button>.

$("input").val("on") : This would check all the radio or check box button whose value is "on".

$("select").val("Orange") : This would select Orange option in a dropdown box with options Orange, Mango and Banana.

$("select").val("Orange", "Mango") : This would select Orange and Mango options in a dropdown box with options Orange, Mango and Banana.

Ask a doubt *(Misuse of 'Ask a Doubt' Section will be dealt as per the Terms & Conditions of Campus Commune)*

Open Doubts | **Closed Doubts**

SHADAB AZHAR

```
<body>
   <div align="right" id="div">this div is left aligned</div>
   <div id="second div">this div is unchanged</div>
   </body>
</html>
```

Above "right " must be written in order to get required answer

about 14 days ago

PARIMALA PITCHIKA

Can a non-IT student get all these things by reading?

about 1 month ago

LAKSHMINADHA PRASAD THOTA

Yes you will get all these concepts by reading..if you have any doubts please share your doubts so that we are ready to answer your questions..

ARUNA K

Being a non-IT student,its really hard to understand those concepts just by reading.Hope we get any practical training during ILP.

10.6. jQuery DOM Traversing

jQuery is a very powerful tool which provides a variety of DOM traversal methods to help us select elements in a document randomly as well as in sequential method.

Most of the DOM Traversal Methods do not modify the jQuery object and they are used to filter out elements from a document based on given conditions.

Find Elements by index:

Consider a simple document with the following HTML content:

```
<html>
<head>
<title>List index</title>
</head>
<body>
<div>
<ul>
    <li>Anil</li>
    <li>Amitabh</li>
    <li>Abhishek</li>
    <li>Shammi</li>
    <li>Jitendra</li>
</ul>
</div>
</body>
</html>
```

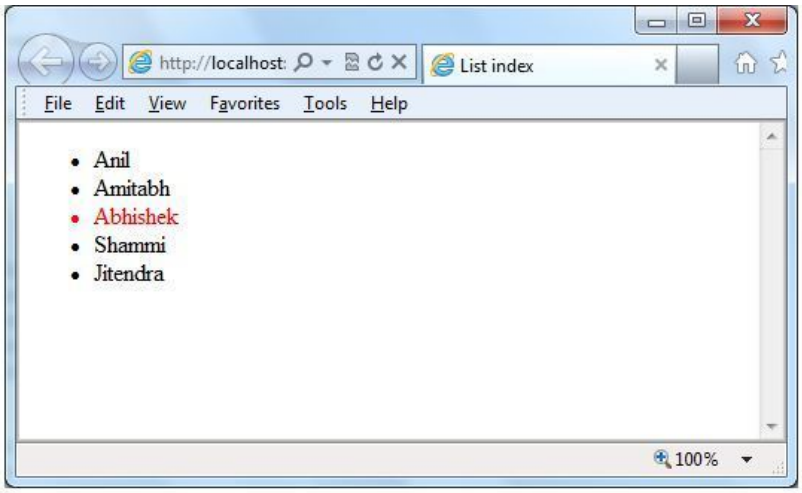
Above every list has its own index, and can be located directly by using eq(index) method as below example.

Every child element starts its index from zero, thus, second list item would be accessed by using $("li").eq(1) and so on.

Example:
Following is a simple example which adds the color to the third list item.

```
<html>
<head>
<title>List index</title>
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript" language="javascript">
    $(document).ready(function() {
        $("li").eq(2).addClass("selected");
    });
</script>
<style>
.selected {
    color: red;
}
</style>

</head>
<body>
<div>
<ul>
    <li>Anil</li>
    <li>Amitabh</li>
    <li>Abhishek</li>
    <li>Shammi</li>
    <li>Jitendra</li>
</ul>
</div>
</body>
</html>
```

The output appears as below:



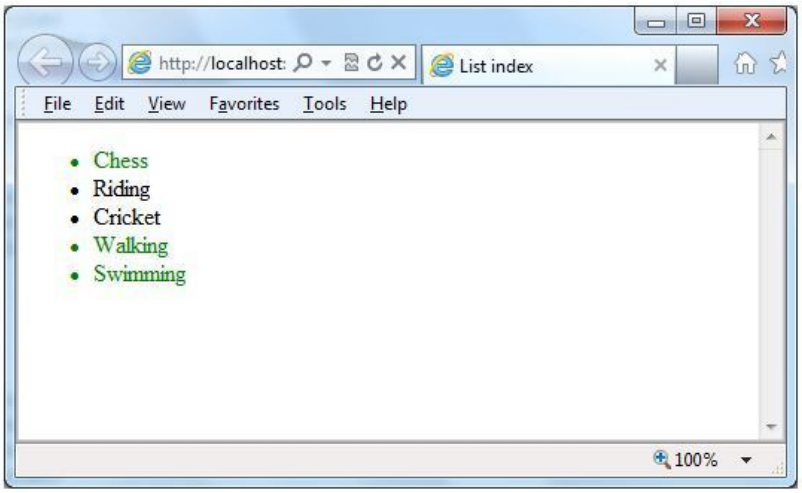
Filtering out Elements:

The filter(selector) method can be used to filter out all elements from the set of matched elements that do not match the specified selector(s). The selector can be written using any selector syntax.

Example:
Following is an example which applies green color to the lists associated with particular class:

```
<html>
<head>
<title>List index</title>
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript" language="javascript">
    $(document).ready(function() {
        $("li").filter(".imp").addClass("selected");
    });
</script>
<style>
.selected {
    color: green;
}
</style>

</head>
<body>
<div>
<ul>
    <li class="imp">Chess</li>
    <li>Riding</li>
    <li>Cricket</li>
    <li class="imp">Walking</li>
    <li class="imp">Swimming</li>
</ul>
</div>
</body>
</html>
```

The result page appears as below:



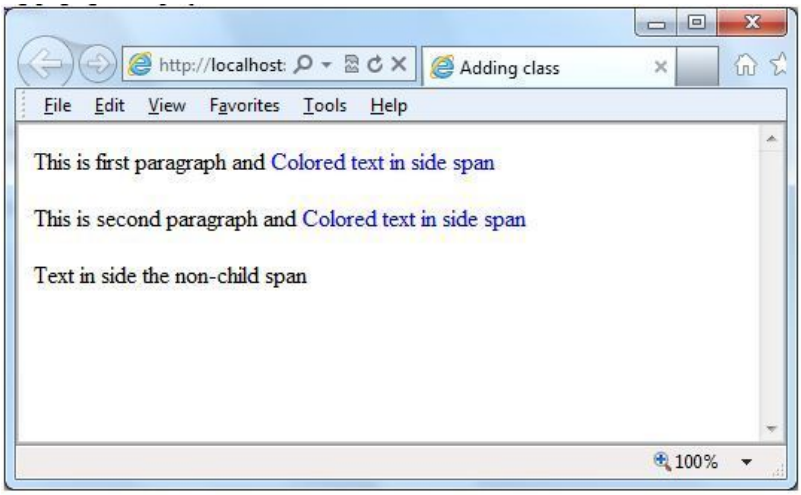
Locating Descendent Elements :

The find(selector) method can be used to locate all the descendent elements of a particular type of elements. The selector can be written using any selector syntax.

Example:
Following is an example which selects all the elements available inside different <p> elements:

```
<html>
<head>
<title>Adding class</title>
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript" language="javascript">
    $(document).ready(function() {
        $("p").find("span").addClass("colored");
    });
</script>
<style>
.colored {
    color: blue;
}
</style>
</head>
<body>
    <p>This is first paragraph and <span>Colored text in side span</span></p>
    <p>This is second paragraph and <span>Colored text in side span</span></p>
    <div>
        <span>Text in side the non-child span</span>
    </div>

</body>
</html>
```

The resulting page gets displayed as below in the web browser:



jQuery DOM Traversing Methods:

Following table lists down useful methods which you can use to filter out various elements from a list of DOM elements:

Selector	Description
eq(index)	Reduce the set of matched elements to a single element.
filter(selector)	Removes all elements from the set of matched elements that do not match the specified selector(s).
filter(fn)	Removes all elements from the set of matched elements that do not match the specified function.
is(selector)	Checks the current selection against an expression and returns true, if at least one element of the selection fits the given selector.
map(callback)	Translate a set of elements in the jQuery object into another set of values in a jQuery array (which may, or may not contain elements).
not(selector)	Removes elements matching the specified selector from the set of matched elements.
slice(start, [end])	Selects a subset of the matched elements.

Following table lists down other useful methods which you can use to locate various elements in a DOM:

Selector	Description
add(selector)	Adds more elements, matched by the given selector, to the set of matched elements.
andSelf( )	Add the previous selection to the current selection.
children([selector])	Get a set of elements containing all of the unique immediate children of each of the matched set of elements.
closest(selector)	Get a set of elements containing the closest parent element that matches the specified selector, the starting element included.
contents( )	Find all the child nodes inside the matched elements (including text nodes), or the content document, if the element is an iframe.
end( )	Revert the most recent 'destructive' operation, changing the set of matched elements to its previous state.
find(selector)	Searches for descendent elements that match the specified selectors.
next([selector])	Get a set of elements containing the unique next siblings of each of the given set of elements.
nextAll([selector])	Find all sibling elements after the current element.
offsetParent( )	Returns a jQuery collection with the positioned parent of the first matched element.
parent([selector])	Get the direct parent of an element. If called on a set of elements, parent returns a set of their unique direct parent elements.
parents([selector])	Get a set of elements containing the unique ancestors of the matched set of elements (except for the root element).
prev([selector])	Get a set of elements containing the unique previous siblings of each of the matched set of elements.
prevAll([selector])	Find all sibling elements in front of the current element.
siblings([selector])	Get a set of elements containing all of the unique siblings of each of the matched set of elements.

Ask a doubt *(Misuse of 'Ask a Doubt' Section will be dealt as per the Terms & Conditions of Campus Commune)*

10.7. jQuery CSS

The jQuery library supports nearly all of the selectors included in Cascading Style Sheet (CSS) specifications 1 through 3, as outlined on the World Wide Web Consortium's site.

Using jQuery library developers can enhance their websites without worrying about browsers and their versions as long as the browsers have JavaScript enabled.

Most of the jQuery CSS Methods do not modify the content of the jQuery object and they are used to apply CSS properties on DOM elements.

Apply CSS Properties:

This is very simple to apply any CSS property using jQuery method css(PropertyName, PropertyValue).

Here is the syntax for the method:

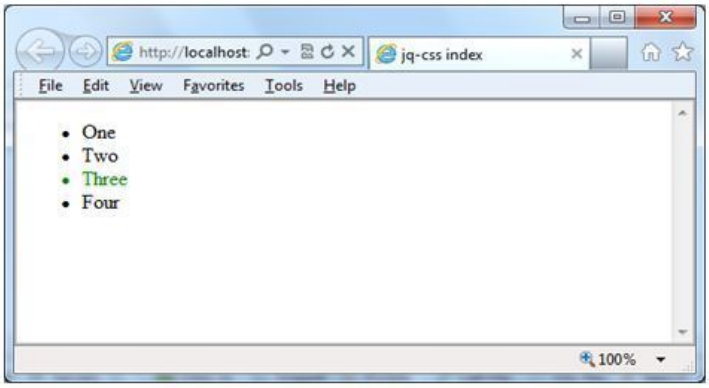
selector.css(PropertyName, PropertyValue);

Here you can pass PropertyName as a javascript string and based on its value, PropertyValue could be string or integer.

Example:
Following is an example which adds font color to the third list item.

```
<html>
<head>
<title>jq-css index</title>
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript" language="javascript">
    $(document).ready(function() {
        $("li").eq(2).css("color","green");
    });
</script>
<style>
.selected {
    color: green;
}
</style>

</head>
<body>
<div>
<ul>
    <li>One</li>
    <li>Two</li>
    <li>Three</li>
    <li>Four</li>
</ul>
</div>
</body>
</html>
```

The results appears as shown below:



Apply Multiple CSS Properties:

You can apply multiple CSS properties using a single jQuery method CSS({key1:val1, key2:val2....). You can apply as many properties as you like in a single call.

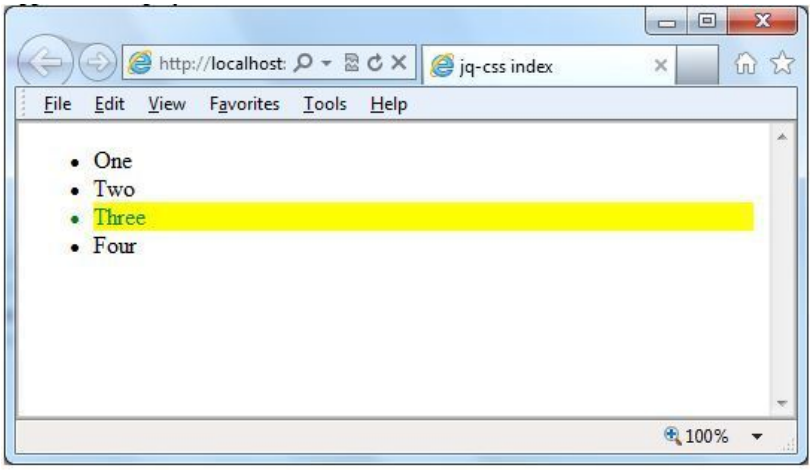
Here is the syntax for the method:
selector.css({key1:val1, key2:val2....keyN:valN})

Here you can pass key as property and val as its value as described above.

Example:
Following is an example which adds font color as well as background color to the third list item.

```
<html>
<head>
<title>jq-css index</title>
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript" language="javascript">
    $(document).ready(function() {
        $("li").eq(2).css({
                "color":"green",
                "background-color":"yellow"
                    });
    });
</script>
<style>
.selected {
    color: green;
}
</style>

</head>
<body>
<div>
<ul>
    <li>One</li>
    <li>Two</li>
    <li>Three</li>
    <li>Four</li>
</ul>
</div>
</body>
</html>
```

The result appears as displayed in below screen:



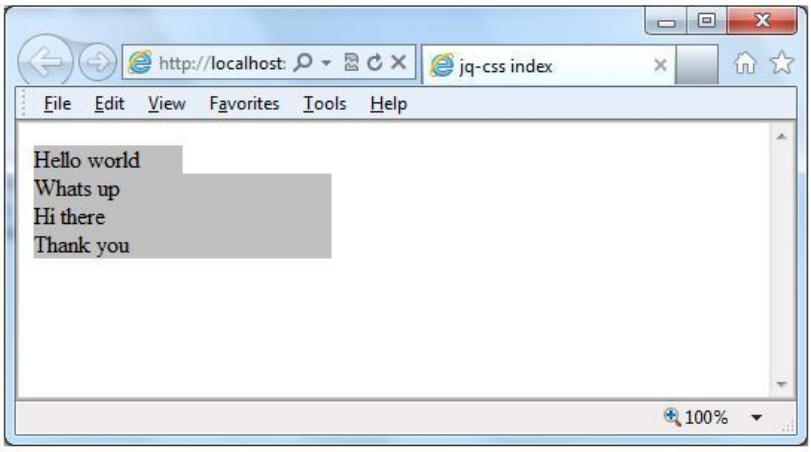
Setting Element Width & Height:

The width(val) and height(val) method can be used to set the width and height respectively of any element.

Example:
Following is a simple example which sets the width of first division element where as rest of the elements have width set by style sheet:

```
<html>
<head>
<title>jq-css index</title>
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript" language="javascript">
    $(document).ready(function() {
        $("div:first").width(100);
    });
</script>
<style>
div {
    width: 200px;
    background: silver;
}
</style>

</head>
<body>
    <div>Hello world</div>
    <div>Whats up</div>
    <div>Hi there</div>
    <div>Thank you</div>
</body>
</html>
```

The result appears as shown in below screen:



jQuery CSS Methods:

Following table lists down all the methods which you can use to play with CSS properties:

Method	Description
css(name)	Return a style property on the first matched element.
css(name, value)	Set a single style property to a value on all matched elements.
css(properties)	Set a key/value object as style properties to all matched elements.
height(val)	Set the CSS height of every matched element.
height()	Get the current computed, pixel, height of the first matched element.
innerHeight()	Gets the inner height (excludes the border and includes the padding) for the first matched element.
innerWidth()	Gets the inner width (excludes the border and includes the padding) for the first matched element.
offset()	Get the current offset of the first matched element, in pixels, relative to the document
offsetParent()	Returns a jQuery collection with the positioned parent of the first matched element.
outerHeight([margin])	Gets the outer height (includes the border and padding by default) for the first matched element.
outerWidth([margin])	Get the outer width (includes the border and padding by default) for the first matched element.
position()	Gets the top and left position of an element relative to its offset parent.
scrollLeft(val)	When a value is passed in, the scroll left offset is set to that value on all matched elements.
scrollLeft()	Gets the scroll left offset of the first matched element.
scrollTop(val)	When a value is passed in, the scroll top offset is set to that value on all matched elements.
scrollTop()	Gets the scroll top offset of the first matched element.
width(val)	Set the CSS width of every matched

Ask a doubt *(Misuse of 'Ask a Doubt' Section will be dealt as per the Terms & Conditions of Campus Commune)*

10.8. jQuery DOM

jQuery provides methods to manipulate DOM in efficient way. You do not need to write big code to modify the value of any element's attribute or to extract HTML code from a paragraph or division.

jQuery provides methods such as .attr(), .html(), and .val() which act as getters, retrieving information from DOM elements for later use.

Content Manipulation:
The html( ) method gets the html contents (innerHTML) of the first matched element.

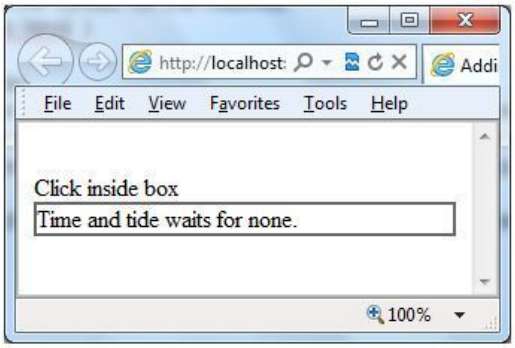
Here is the syntax for the method:
selector.html( )

Example:

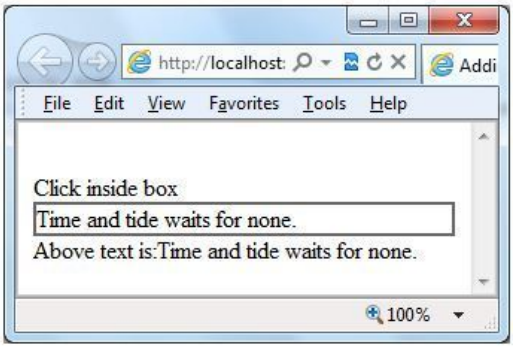
Following is an example which makes use of .html() and .text(val) methods. Here .html() retrieves HTML content from the object and then .text(val) method sets value of the object using passed parameter:

```
<html>
<head>
<title>Adding text</title>
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript" language="javascript">
    $(document).ready(function() {
        $("#division").click(function() {
            var content = $(this).html();
            $("#result").text("Above text is:"+content);
        });
    });
</script>

</head>
<body>
<br/>Click inside box
<div id="division" style="border: 2px solid #666;">
    Time and tide waits for none.
</div>
<div id="result"></div>
</body>
</html>
```

When loaded the page appears as below:



When clicked in rectangle div area; the result is as shown below:



DOM Element Replacement:

You can replace a complete DOM element with the specified HTML or DOM elements. The replaceWith(content) method serves this purpose very well.

Here is the syntax for the method:

selector.replaceWith(content)

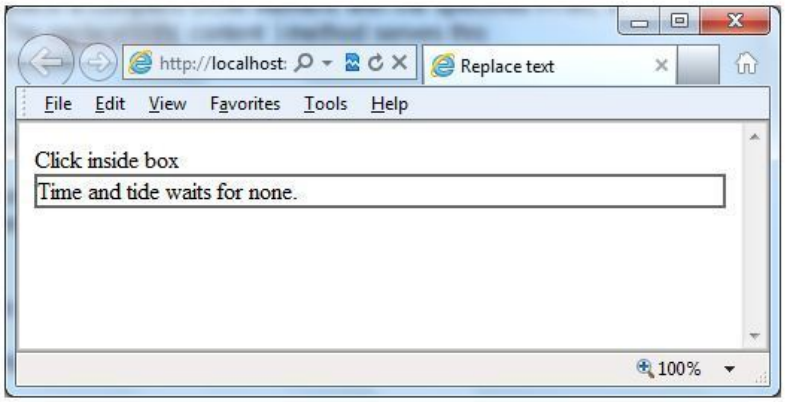
Here content is what you want to have instead of original element. This could be HTML or simple text.

Example:

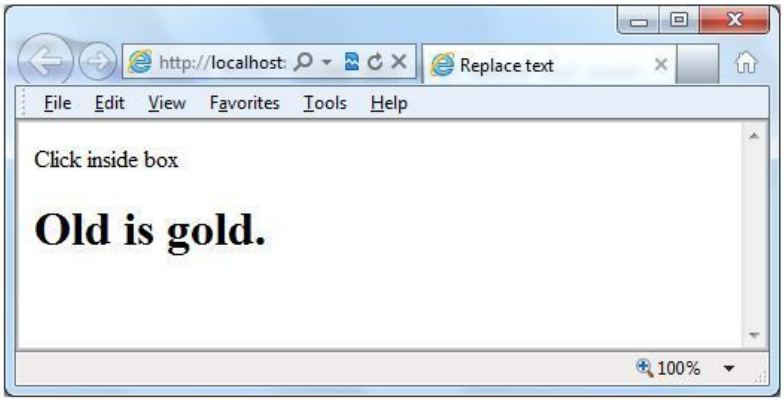
Following is an example which would replace division element with <h1>:

```
<html>
<head>
<title>Replace text</title>
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript" language="javascript">
    $(document).ready(function() {
        $("div").click(function() {
            $(this).replaceWith("<h1>Old is gold.</h1>");
        })
    });
</script>

</head>

<body>
Click inside box<br/>
<div style="border: 2px solid #666;">
Time and tide waits for none.
</div>
</body>
</html>
```

When loaded; page is displayed as below:



After clicking, you see the below result:



Removing DOM Elements:

There may be a situation when you would like to remove one or more DOM elements from the document. jQuery provides two methods to handle the situation.

The empty( ) method remove all child nodes from the set of matched elements where as the method remove(expr) method removes all matched elements from the DOM.

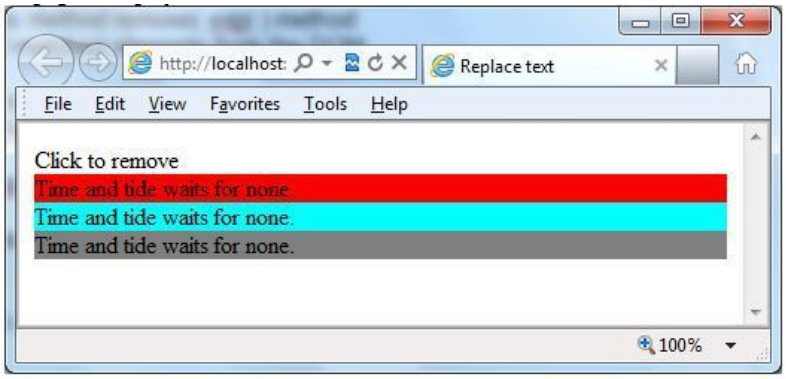
Here is the syntax for the method:
selector.remove([expr])
or
selector.empty( )

You can pass optional parameter expr to filter the set of elements to be removed.

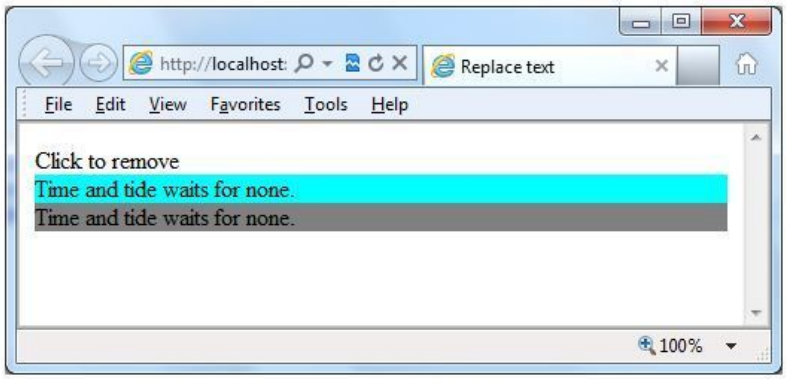
Example:
Following is an example where elements are being removed as soon as they are clicked:

```
<html>
<head>
<title>Replace text</title>
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript" language="javascript">
    $(document).ready(function() {
        $("div").click(function() {
            $(this).remove();
        })
    });
</script>
</head>
<body>
Click to remove<br/>
<div style="background: red;">Time and tide waits for none.</div>
<div style="background: aqua">Time and tide waits for none.</div>
<div style="background: gray;">Time and tide waits for none.</div>
</body>
</html>
```

When loaded; page is displayed as below:



On clicking one of the divs; it gets removed as shown in below result:



Inserting DOM elements:

There may be a situation when you would like to insert new one or more DOM elements in your existing document. jQuery provides various methods to insert elements at various locations.

The after(content) method insert content after each of the matched elements where as the method before(content) method inserts content before each of the matched elements.

Here is the syntax for the method:
selector.after(content)
or
selector.before(content)

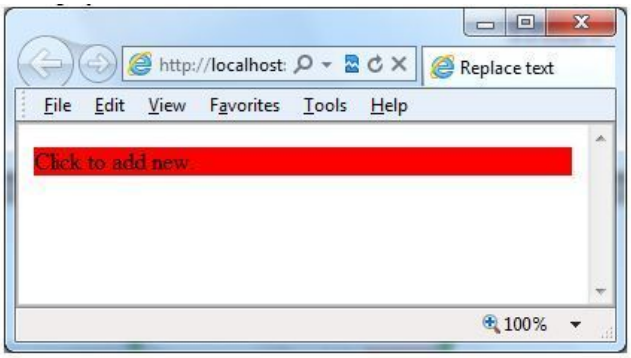
Here content is what you want to insert. This could be HTML or simple text.

Example:

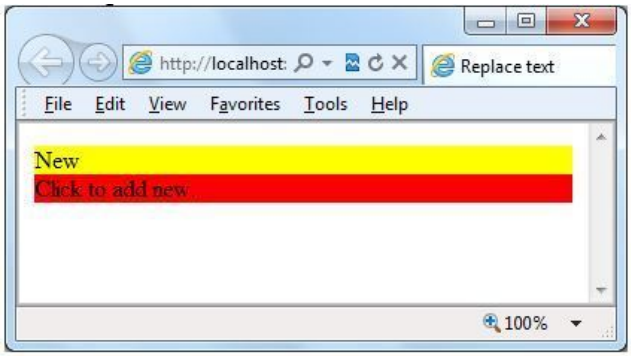
Following is an example where elements are being inserted just before the clicked element:

```
<html>
<head>
<title>Replace text</title>
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript" language="javascript">
    $(document).ready(function() {
        $("div").click(function() {
            $(this).before("<div style='background: yellow;'>New</div>");
        })
    });
</script>
</head>
<body>
<div style="background: red;">Click to add new.</div>
</body>
</html>
```

When loaded; page is displayed as below:



On clicking; the new element gets added as shown below:



DOM Manipulation Methods:

Following table lists down all the methods which you can use to manipulate DOM elements:

Method	Description
after(content)	Insert content after each of the matched elements.
append(content)	Append content to the inside of every matched element.
appendTo(selector)	Append all of the matched elements to another, specified, set of elements.
before(content)	Insert content before each of the matched elements.
clone(bool)	Clone matched DOM Elements, and all their event handlers, and select the clones.
clone( )	Clone matched DOM Elements and select the clones.
empty( )	Remove all child nodes from the set of matched elements.
html(val)	Set the html contents of every matched element.
html( )	Get the html contents (innerHTML) of the first matched element.
insertAfter(selector)	Insert all of the matched elements after another, specified, set of elements.
insertBefore(selector)	Insert all of the matched elements before another, specified, set of elements.
prepend(content)	Prepends content to the inside of every matched element.
prependTo(selector)	Prepends all of the matched elements to another, specified, set of elements.
remove(expr)	Removes all matched elements from the DOM.
replaceAll(selector)	Replaces the elements matched by the specified selector with the matched elements.
replaceWith(content)	Replaces all matched elements with the specified HTML or DOM elements.
text(val)	Set the text contents of all matched elements.
text( )	Get the combined text contents of all matched elements.
text(val)	Set the text contents of all matched elements.
text( )	Get the combined text contents of all matched elements.
wrap(elem)	Wrap each matched element with the specified element.
wrap(html)	Wrap each matched element with the specified HTML content.
wrapAll(elem)	Wrap all the elements in the matched set into a single wrapper element.
wrapAll(html)	Wrap all the elements in the matched set into a single wrapper element.
wrapInner(elem)	Wrap the inner child contents of each matched element (including text nodes) with a DOM element.
wrapInner(html)	Wrap the inner child contents of each matched element (including text nodes) with an HTML structure.

Ask a doubt *(Misuse of 'Ask a Doubt' Section will be dealt as per the Terms & Conditions of Campus Commune)*

Open Doubts | **Closed Doubts**

KRISHNA KABI

I am having a problem while running the above codes using notepad++. On clicking on the respective webpage no further action is being carried out as per the code. Kindly notify me in case I am wrong. Thank you.

about 1 month ago

SUMAN SINGH

<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.10.2/jquery.min.js">
</script>

Insert this code too in your head

10.9. jQuery Events

We have the ability to create dynamic web pages by using events. Events are actions that can be detected by your Web Application.

Following are the examples events:

- A mouse click
- A web page loading
- Taking mouse over an element
- Submitting an HTML form
- A keystroke on your keyboard etc.

When these events are triggered you can then use a custom function to do pretty much whatever you want with the event. These custom functions call Event Handlers.

Binding event handlers:

Using the jQuery Event Model, we can establish event handlers on DOM elements with the bind() method as follows:

```
$('div').bind('click', function( event ){
  alert('Hi there!');
});
```

This code will cause the division element to respond to the click event; when a user clicks inside this division thereafter, the alert will be shown.

The full syntax of the bind() command is as follows:

selector.bind(eventType[, eventData], handler)

Following is the description of the parameters:

eventType: A string containing a JavaScript event type, such as click or submit. Refer to the next section for a complete list of event types.

eventData: This is optional parameter is a map of data that will be passed to the event handler.

handler: A function to execute each time the event is triggered.

Removing event handlers:

Typically, once an event handler is established, it remains in effect for the remainder of the life of the page. There may be a need when you would like to remove event handler.

jQuery provides the unbind() command to remove an exiting event handler. The syntax of unbind() is as follows:

selector.unbind(eventType, handler)

or

selector.unbind(eventType)

Following is the description of the parameters:

eventType: A string containing a JavaScript event type, such as click or submit. Refer to the next section for a complete list of event types.

handler: If provided, identifies the specific listener that is to be removed.

Event Types:

The following are cross platform and recommended event types which you can bind using jQuery:

Event Type	Description
blur	Occurs when the element loses focus
change	Occurs when the element changes
click	Occurs when a mouse click
dblclick	Occurs when a mouse double-click
error	Occurs when there is an error in loading or unloading etc.
focus	Occurs when the element gets focus
keydown	Occurs when key is pressed
keypress	Occurs when key is pressed and released
keyup	Occurs when key is released
load	Occurs when document is loaded
mousedown	Occurs when mouse button is pressed
mouseenter	Occurs when mouse enters in an element region
mouseleave	Occurs when mouse leaves an element region
mousemove	Occurs when mouse pointer moves
mouseout	Occurs when mouse pointer moves out of an element
mouseover	Occurs when mouse pointer moves over an element

mouseup	Occurs when mouse button is released
resize	Occurs when window is resized
scroll	Occurs when window is scrolled
select	Occurs when a text is selected
submit	Occurs when form is submitted
unload	Occurs when documents is unloaded

The Event Object:

The callback function takes a single parameter; when the handler is called the JavaScript event object will be passed through it.

The event object is often unnecessary and the parameter is omitted, as sufficient context is usually available when the handler is bound to know exactly what needs
to be done when the handler is triggered, however there are certain attributes which you would need to be accessed.

The Event Attributes:

The following event properties/attributes are available and safe to access in a platform independent manner:

Property	Description
altKey	Set to true if the Alt key was pressed when the event was triggered, false if not. The Alt key is labeled Option on most Mac keyboards.
ctrlKey	Set to true if the Ctrl key was pressed when the event was triggered, false if not.
data	The value, if any, passed as the second parameter to the bind() command when the handler was established.
keyCode	For keyup and keydown events, this returns the key that was pressed.
metaKey	Set to true if the Meta key was pressed when the event was triggered, false if not. The Meta key is the Ctrl key on PCs and the Command key on Macs.
pageX	For mouse events, specifies the horizontal coordinate of the event relative from the page origin.
pageY	For mouse events, specifies the vertical coordinate of the event relative from the page origin.
relatedTarget	For some mouse events, identifies the element that the cursor left or entered when the event was triggered.
screenX	For mouse events, specifies the horizontal coordinate of the event relative from the screen origin.
screenY	For mouse events, specifies the vertical coordinate of the event relative from the screen origin.
shiftKey	Set to true if the Shift key was pressed when the event was triggered, false if not.
target	Identifies the element for which the event was triggered.
timeStamp	The timestamp (in milliseconds) when the event was created.
type	For all events, specifies the type of event that was triggered (for example, click).
which	For keyboard events, specifies the numeric code for the key that caused the event, and for mouse events, specifies which button

The Event Methods:

There is a list of methods which can be called on an Event Object:

Method	Description
preventDefault()	Prevents the browser from executing the default action.
isDefaultPrevented()	Returns whether event.preventDefault() was ever called on this event object.
stopPropagation()	Stops the bubbling of an event to parent elements, preventing any parent handlers from being notified of the event.
isPropagationStopped()	Returns whether event.stopPropagation() was ever called on this event object.
stopImmediatePropagation()	Stops the rest of the handlers from being executed.
isImmediatePropagationStopped()	Returns whether event.stopImmediatePropagation() was ever called on this event object.

Event Manipulation Methods:

Following table lists down important event-related methods:

Method	Description
bind(type, [data], fn)	Binds a handler to one or more events (like click) for each matched element. Can also bind custom events.
die(type, fn)	This does the opposite of live, it removes a bound live event.
hover(over, out)	Simulates hovering for example moving the mouse on, and off, an object.
live(type, fn)	Binds a handler to an event (like click) for all current - and future - matched element. Can also bind custom events.
one(type, [data], fn)	Binds a handler to one or more events to be executed once for each matched element.
ready(fn)	Binds a function to be executed whenever the DOM is ready to be traversed and manipulated.
toggle(fn, fn2, fn3,...)	Toggle among two or more function calls every other click.
trigger(event, [data])	Trigger an event on every matched element.
triggerHandler(event, [data])	Triggers all bound event handlers on an element.
unbind([type], [fn])	This does the opposite of bind, it removes bound events from each of the matched elements.

Event Helper Methods:

jQuery also provides a set of event helper functions which can be used either to trigger an event to bind any event types mentioned above.

Trigger Methods:

Following is an example which would triggers the blur event on all paragraphs:

```
$("p").blur();
```

Binding Methods:

Following is an example which would bind a click event on all the :

```
$("div").click( function () {
  // do something here
});
```

Here is a complete list of all the support methods provided by jQuery:

Method	Description
blur( )	Triggers the blur event of each matched element.
blur(fn)	Bind a function to the blur event of each matched element.
change( )	Triggers the change event of each matched element.
change(fn)	Binds a function to the change event of each matched element.
click( )	Triggers the click event of each matched element.
click(fn)	Binds a function to the click event of each matched element.
dblclick( )	Triggers the dblclick event of each matched element.
dblclick(fn)	Binds a function to the dblclick event of each matched element.
error( )	Triggers the error event of each matched element.
error(fn)	Binds a function to the error event of each matched element.
focus( )	Triggers the focus event of each matched element.
focus(fn)	Binds a function to the focus event of each matched element.
keydown( )	Triggers the keydown event of each matched element.
keydown(fn)	Bind a function to the keydown event of each matched element.
keypress( )	Triggers the keypress event of each matched element.
keypress(fn)	Binds a function to the keypress event of each matched element.
keyup( )	Triggers the keyup event of each matched element.
keyup(fn)	Bind a function to the keyup event of each matched element.
load(fn)	Binds a function to the load event of each matched element.
mousedown(fn)	Binds a function to the mousedown event of each matched element.
mouseenter(fn)	Bind a function to the mouseenter event of each matched element.
mouseleave(fn)	Bind a function to the mouseleave event of each matched element.
mousemove(fn)	Bind a function to the mousemove event of each matched element.
mouseout(fn)	Bind a function to the mouseout event of each matched element.
mouseover(fn)	Bind a function to the mouseover event of each matched element.
mouseup(fn)	Bind a function to the mouseup event of each matched element.
resize(fn)	Bind a function to the resize event of each matched element.

scroll(fn)	Bind a function to the scroll event of each matched element.
select( )	Trigger the select event of each matched element.
select(fn)	Bind a function to the select event of each matched element.
submit( )	Trigger the submit event of each matched element.
submit(fn)	Bind a function to the submit event of each matched element.
unload(fn)	Binds a function to the unload event of each matched element.

Ask a doubt *(Misuse of 'Ask a Doubt' Section will be dealt as per the Terms & Conditions of Campus Commune)*

10.10. jQuery Effects

jQuery provides a trivially simple interface for doing various kind of amazing effects. jQuery methods allow us to quickly apply commonly used effects with a minimum configuration.

This tutorial covers all the important jQuery methods to create visual effects.

Showing and Hiding elements:
The commands for showing and hiding elements are pretty much what we would expect: show() to show the elements in a wrapped set and hide() to hide them.

Syntax:
Here is the simple syntax for show() method:

[selector].show(speed, [callback]);

Here is the description of all the parameters:
speed: A string representing one of the three predefined speeds ("slow", "normal", or "fast") or the number of milliseconds to run the animation (e.g. 1000).
callback: This optional parameter represents a function to be executed whenever the animation completes; executes once for each element animated against.

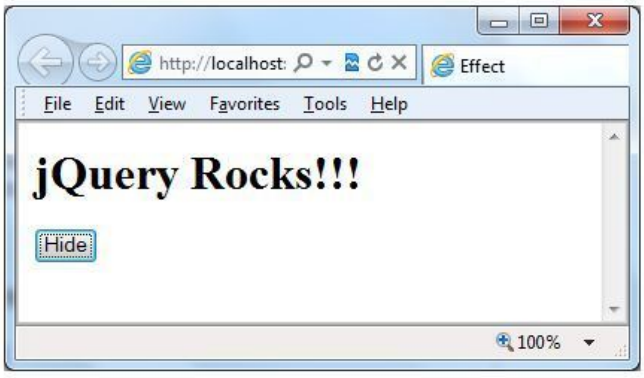
Following is the simple syntax for hide() method:
[selector].hide(speed, [callback]);

Here is the description of all the parameters:
speed: A string representing one of the three predefined speeds ("slow", "normal", or "fast") or the number of milliseconds to run the animation (e.g. 1000).
callback: This optional parameter represents a function to be executed whenever the animation completes; executes once for each element animated against.

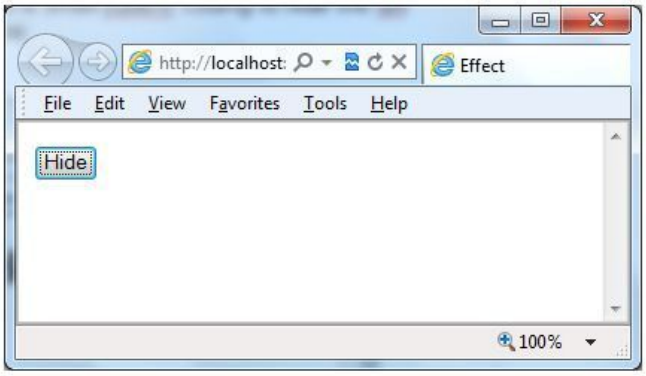
Example:
Consider the following HTML file with jQuery coding to hide the div:

```
<html>
<head>
<title>Effect</title>
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript" language="javascript">
    $(document).ready(function() {
        $("#hide").click(function() {
            $(".mydiv").hide(1000);
        });
    });
</script>
<style>
</style>
</head>
<body>
<div class="mydiv"><h1>jQuery Rocks!!!</h1></div>
<input id="hide" type="button" value="Hide" />
</body>
</html>
```

When loaded; page appears as below:



On clicking the Hide button the div disappears as seen below:



Toggling the elements:

jQuery provides methods to toggle the display state of elements between revealed or hidden. If the element is initially displayed, it will be hidden; if hidden, it will be
shown.

Syntax:

Here is the simple syntax for one of the toggle() methods:

[selector]..toggle([speed][, callback]);

Here is the description of all the parameters:

speed: A string representing one of the three predefined speeds ("slow", "normal", or "fast") or the number of milliseconds to run the animation (e.g. 1000).

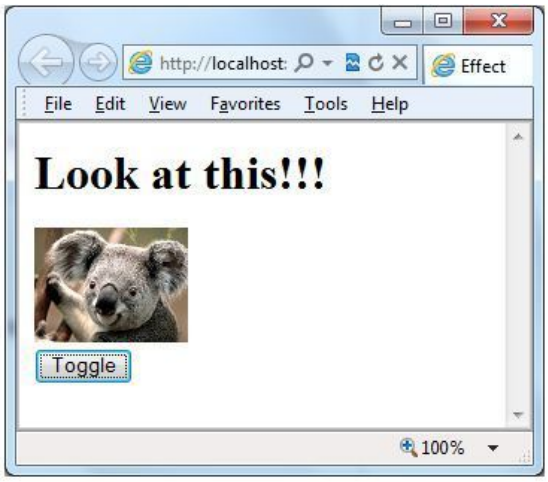
callback: This optional parameter represents a function to be executed whenever the animation completes; executes once for each element animated against.

Example:

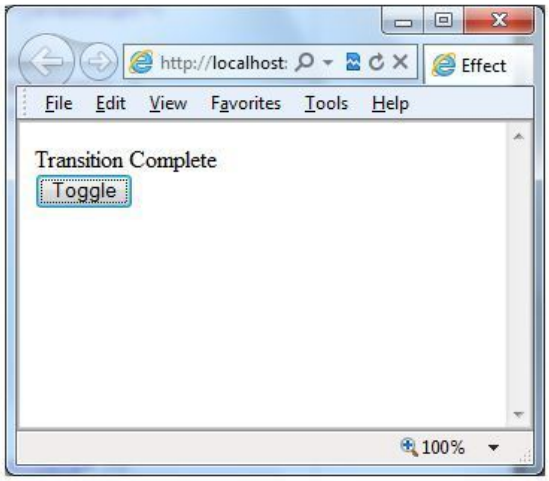
We can animate any element, such as a containing an image/text:

```
<html>
<head>
<title>Effect</title>
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript" language="javascript">
    $(document).ready(function() {
        $("#button").click(function(event) {
            $(".mydiv").toggle('slow', function() {
                $(".log").text('Transition Complete');
            });
        });
    });
</script>
<style>
</style>
</head>
<body>
<div class="mydiv">
<h1>Look at this!!!</h1>
<img src="pic.jpg"/>
</div>
<div class="log"></div>
<input id="button" type="button" value="Toggle" />
</body>
</html>
```

When loaded; page appears as below:



On clicking the button the content disappears and message is displayed as below:



jQuery Effect Methods:

You have seen basic concept of jQuery Effects. Following table lists down all the important methods to create different kind of effects:

Methods	Description
animate(params, [duration, easing, callback])	A function for making custom animations.
fadeIn(speed, [callback])	Fade in all matched elements by adjusting their opacity and firing an optional callback after completion.
fadeOut(speed, [callback])	Fade out all matched elements by adjusting their opacity to 0, then setting display to "none" and firing an optional callback after completion.
fadeTo(speed, opacity, callback)	Fade the opacity of all matched elements to a specified opacity and firing an optional callback after completion.
hide( )	Hides each of the set of matched elements if they are shown.
hide(speed, [callback])	Hide all matched elements using a graceful animation and firing an optional callback after completion.
show( )	Displays each of the set of matched elements if they are hidden.
show(speed, [callback])	Show all matched elements using a graceful animation and firing an optional callback after completion.

slideDown(speed, [callback])	Reveal all matched elements by adjusting their height and firing an optional callback after completion.
slideToggle(speed, [callback])	Toggle the visibility of all matched elements by adjusting their height and firing an optional callback after completion.
slideUp(speed, [callback])	Hide all matched elements by adjusting their height and firing an optional callback after completion.
stop([clearQueue, gotoEnd])	Stops all the currently running animations on all the specified elements.
toggle( )	Toggle displaying each of the set of matched elements.
toggle(speed, [callback])	Toggle displaying each of the set of matched elements using a graceful animation and firing an optional callback after completion.
toggle(switch)	Toggle displaying each of the set of matched elements based upon the switch (true shows all elements, false hides all elements).

UI Library Based Effects:

To use these effects you would have to download jQuery UI Library jquery-ui-1.7.2.custom.min.js or latest version of this UI library from jQuery UI Library (http://jqueryui.com/download).

After extracting jquery-ui-1.7.2.custom.min.js file from the download, you would include this file in similar way as you include core jQuery Library file.

Methods	Description
Blind	Blinds the element away or shows it by blinding it in.
Bounce	Bounces the element vertically or horizontally n-times.
Clip	Clips the element on or off, vertically or horizontally.
Drop	Drops the element away or shows it by dropping it in.
Explode	Explodes the element into multiple pieces.
Fold	Folds the element like a piece of paper.
Highlight	Highlights the background with a defined color.
Puff	Scale and fade out animations create the puff effect.
Pulsate	Pulsates the opacity of the element multiple times.
Scale	Shrink or grow an element by a percentage factor.
Shake	Shakes the element vertically or horizontally n-times.
Size	Resize an element to a specified width and height.
Slide	Slides the element out of the viewport.
Transfer	Transfers the outline of an element to another.

10.11. Practice Problems

Javascript problems:

1. Apply below validations to Add doctor page created in HTML and CSS exercise:

i. Display appropriate message in case a mandatory field is left blank or no value is specified.
ii. First and last name cannot be greater than 20 characters. Allow only alphabets.
iii. contact number should exactly have 10 digits
iv. email id should have an '@' and a '.')
v. NoOfPatientsPerDay and DepartmentNo should be only numbers

2. Apply below validations to Add department page created in HTML and CSS exercise:

i. Department name cannot be greater than 30 characters, no special characters.
ii. Number of doctors should be only numbers.
iii. Department description should be of maximum of 50 characters.
iv. Cost of each bed, Cost of each Cabin, Cost of each ICU should be only decimal numbers up to 2 decimal places.

3. Apply below validations to Add patient page created in HTML and CSS exercise:

i. Patient name,Father/spouse name cannot be greater than 30 characters, Allow only alphabets.
ii. ContactNumber sould be of 10 digits.
iii. email id should have an '@' and a '.')

Jquery problems:

1. Create an HTML page and on click of button select an option in a Select Box.

2. Create an HTML page and on click of button change size of div to double.

3. Create an HTML page and on click of button remove all the child elements from div.

4. Create an HTML page and on click of button show (in alert) the number of children of a div tag

5. Create an HTML page and on click of button change all the h2 tag to h3

6. Create an HTML page which contains different divs/paragraphs of text with one buttons. On click of button make the word named 'TCS' appear as bold in the page.

7. Create an HTML page which contains a table of size 3x3. Table cells contains numbers in it. On click of button change color of every cell which has number greater than 10. When mouse is hovered on the cell, change the color to red if the cell does not contain numeric value

Click here to download the solutions

Ask a doubt *(Misuse of 'Ask a Doubt' Section will be dealt as per the Terms & Conditions of Campus Commune)*