Questions Answer 2073

1.

(a) Write the differences between client side and server side scripting language. – 5

Answer:
Difference between client side scripting and server side scripting :

Client side scripting	Server side scripting
It runs on user's computer.	It runs on web server.

i. Client side scripting :
Web browsers execute client side scripting. It is use when browsers has all code. Source code used to transfer from web server to user's computer over internet and run directly on browsers. It is also used for validations and functionality for user events.

It allows for more interactivity. It usually performs several actions without going to user. It cannot be basically used to connect to databases on web server. These scripts cannot access file system that resides at web browser. Pages are altered on basis of users choice. It can also used to create "cookies" that store data on user's computer.

ii. Server side scripting :
Web servers are used to execute server side scripting. They are basically used to create dynamic pages. It can also access the file system residing at web server. Server-side environment that runs on a scripting language is a web-server.
Scripts can be written in any of a number of server-side scripting language available. It is used to retrieve and generate content for dynamic pages. It is used to require to download plugins. In this load times are generally faster than client-side scripting. When you need to store and retrieve information a database will be used to contain data. It can use huge resources of server. It reduces client-side computation overhead. Server sends pages to request of user/client.

Client side scripting	Server side scripting
Source code is visible to user.	Source code is not visible to user because it's output of server side is a HTML page.
It usually depends on browser and it's version.	In this any server side technology can be use and it does not depend on client.
It runs on user's computer.	It runs on web server.

Client side scripting	Server side scripting
There are many advantages link with this like faster. response times, a more interactive application.	The primary advantage is it's ability to highly customize, response requirements, access rights based on user.
It does not provide security for data.	It provides more security for data.
It is a technique use in web development in which scripts runs on clients browser.	It is a technique that uses scripts on web server to produce a response that is customized for each clients request.
HTML, CSS and javascript are used.	PHP, Python, Java, Ruby are used.

(b) Write advantage and disadvantage of JavaScript. 3

Answer:
Advantages of JavaScript

- **Speed -** JavaScript tends to be very fast because it is often run immediately within the client's browser. So long as it doesn't require outside resources, JavaScript isn't slowed down by calls to a backend server. Also, major browsers all support JIT (just in time) compilation for JavaScript, meaning that there's no need to compile the code before running it.

- **Simplicity -** JavaScript's syntax was inspired by Java's and is relatively easy to learn compared to other popular languages like C++.

- **Popularity -** JavaScript is everywhere on the web, and with the advent of Node.js, is increasingly used on the backend. There are countless resources to learn JavaScript. Both StackOverflow and GitHub show an increasing amount of projects that use JavaScript, and the traction it's gained in recent years is only expected to increase.

- **Interoperability -** Unlike PHP or other scripting languages, JavaScript can be inserted into any web page. JavaScript can be used in many different kinds of applications because of support in other languages like Pearl and PHP.

- **Server Load -** JavaScript is client-side, so it reduces the demand on servers overall, and simple applications may not need a server at all.

- **Rich interfaces -** JavaScript can be used to create features like drag and drop and components such as sliders, all of which greatly enhance the user interface and experience of a site.
- **Extended Functionality -** Developers can extend the functionality of web pages by writing snippets of JavaScript for third party add-ons like Grease monkey.

- **Versatility** - There are many ways to use JavaScript through Node.js servers. If you were to bootstrap Node.js with Express, use a document database like MongoDB, and use JavaScript on the frontend for clients, it is possible to develop an entire JavaScript app from front to back using only JavaScript.

- **Updates** - Since the advent of ECMAScript 5 (the scripting specification that JavaScript relies on), ECMA International has been dedicated to updating JavaScript annually. So far, we have received browser support for ES6 in 2017 and look forward to ES7 being supported in the future.

Disadvantages of JavaScript

- **Client-Side Security** - Since JavaScript code is executed on the client-side, bugs and oversights can sometimes be exploited for malicious purposes. Because of this, some people choose to disable JavaScript entirely.

- **Browser Support** - While server-side scripts always produce the same output, different browsers sometimes interpret JavaScript code differently. These days the differences are minimal, and you shouldn't have to worry about it as long as you test your script in all major browsers.

2.

(a) Define JavaScript data type with example. 4

Answer:

JavaScript Data Types

In this tutorial, you will learn about the various data types available in JavaScript with the help of examples.

There are different types of data that we can use in a JavaScript program. For example,

```
const x = 5;
const y = "Hello";
```

Here,

- 5 is an integer data.
- "Hello" is a string data.

JavaScript Data Types

There are eight basic data types in JavaScript. They are:

Data Types	Description	Example
String	represents textual data	'hello', "hello world!" etc
Number	an integer or a floating-point number	3, 3.234, 3e-2 etc.
BigInt	an integer with arbitrary precision	900719925124740999n , 1n etc.
Boolean	Any of two values: true or false	true and false
undefined	a data type whose variable is not	let a;

	initialized	
null	denotes a null value	let a = null;
Symbol	data type whose instances are unique and immutable	let value = Symbol('hello');
Object	key-value pairs of collection of data	let student = { };

Here, all data types except Object are primitive data types, whereas Object is non-primitive.

Note: The Object data type (non-primitive type) can store collections of data, whereas primitive data type can only store a single data.

(c) Explain various looping used in JavaScript. 6

Answer:

What is looping in JavaScript?

JavaScript Loops are **used to repeatedly run a block of code** - until a certain condition is met. When developers talk about iteration or iterating over, say, an array, it is the same as looping.

How many types of loops are there in JavaScript?

JavaScript now supports **five different types** of loops: while — loops through a block of code as long as the condition specified evaluates to true. do… while — loops through a block of code once; then the condition is evaluated.

JavaScript Loops

Loops are handy, if you want to run the same code over and over again, each time with a different value.

Often this is the case when working with arrays:

Instead of writing:

```
text += cars[0] + "<br>";
text += cars[1] + "<br>";
text += cars[2] + "<br>";
text += cars[3] + "<br>";
text += cars[4] + "<br>";
text += cars[5] + "<br>";
```

You can write:

```
for (let i = 0; i < cars.length; i++) {
  text += cars[i] + "<br>";
}
```

Different Kinds of Loops

JavaScript supports different kinds of loops:

- for - loops through a block of code a number of times
- for/in - loops through the properties of an object
- for/of - loops through the values of an alterable object
- while - loops through a block of code while a specified condition is true
- do/while - also loops through a block of code while a specified condition is true

The For Loop
The for loop has the following syntax:

```
for (statement 1; statement 2; statement 3) {
  // code block to be executed
}
```

Statement 1 is executed (one time) before the execution of the code block.
Statement 2 defines the condition for executing the code block.
Statement 3 is executed (every time) after the code block has been executed.

Example

```
for (let i = 0; i < 5; i++) {
  text += "The number is " + i + "<br>";
}
```

From the example above, you can read:
Statement 1 sets a variable before the loop starts (let i = 0).
Statement 2 defines the condition for the loop to run (i must be less than 5).
Statement 3 increases a value (i++) each time the code block in the loop has been executed.
Statement 1

Normally you will use statement 1 to initialize the variable used in the loop (let i = 0).
This is not always the case, JavaScript doesn't care. Statement 1 is optional.
You can initiate many values in statement 1 (separated by comma):

Example

```
for (let i = 0, len = cars.length, text = ""; i < len; i++) {
  text += cars[i] + "<br>";
}
```

Example

```
let i = 2;
let len = cars.length;
let text = "";
for (; i < len; i++) {
  text += cars[i] + "<br>";
}
```

Statement 2
Often statement 2 is used to evaluate the condition of the initial variable.
This is not always the case, JavaScript doesn't care. Statement 2 is also optional.
If statement 2 returns true, the loop will start over again, if it returns false, the loop will end.
If you omit statement 2, you must provide a **break** inside the loop. Otherwise the loop will never end. This will crash your browser. Read about breaks in a later chapter of this tutorial.

Statement 3
Often statement 3 increments the value of the initial variable.
This is not always the case, JavaScript doesn't care, and statement 3 is optional.
Statement 3 can do anything like negative increment (i--), positive increment (i = i + 15), or anything else.
Statement 3 can also be omitted (like when you increment your values inside the loop):

Example

```
let i = 0;
let len = cars.length;
```

```
let text = "";
for (; i < len; ) {
  text += cars[i] + "<br>";
  i++;
}
```

Loop Scope
Using var in a loop:

Example

```
var i = 5;

for (var i = 0; i < 10; i++) {
  // some code
}

// Here i is 10
```

Using let in a loop:

Example

```
let i = 5;

for (let i = 0; i < 10; i++) {
  // some code
}

// Here i is 5
```

In the first example, using var, the variable declared in the loop redeclares the variable outside the loop.
In the second example, using let, the variable declared in the loop does not redeclare the variable outside the loop.
When let is used to declare the i variable in a loop, the i variable will only be visible within the loop.

For/Of and For/In Loops
The for/in loop and the for/of loop are explained in the next chapter.

While Loops
The while loop and the do/while are explained in the next chapters.

Exercise:
Create a loop that runs from 0 to 9.

```
let i;
[ ] ([ ] = [ ]; [ ] < [ ]; [ ]) {
  console.log(i);
}
```

The For In Loop
The JavaScript for in statement loops through the properties of an Object:

Syntax

```
for (key in object) {
  // code block to be executed
}
```

Example

```
const person = {fname:"John", lname:"Doe", age:25};

let text = "";
for (let x in person) {
  text += person[x];
}
```

3.

(a) Describe inheritance. 3

Answer:

What best describes the inheritance?

1. Which among the following best describes the Inheritance? Explanation: It can only be indicated by using the **data and functions** that we use in derived class, being provided by parent class. ... Explanation: A class inheriting a base class defines single level inheritance.

(b) Explain sub procedures with example. – 9

Answer: What is the importance of sub procedure?

Sub procedures **help to make programs smaller and seamless to manage**. A sub procedure begins with a Sub keyword and ends with an End Sub keyword. The parameter is a certain data that is passed into the sub procedure to perform a specified task.

What is sub procedure explain with an example?

A Sub procedure is **a series of Visual Basic statements enclosed by the Sub and End Sub statements**. The Sub procedure performs a task and then returns control to the calling code, but it does not return a value to the calling code. ... You can define a Sub procedure in modules, classes, and structures.

Difference between sub-procedure and function

Functions or Sub Procedures ?

In programming languages Functions and Procedure are the code segments written separately to take advantage of reusability. If you want write the same code segments many places in the programming file, it is likely that your program would benefit from Functions or Procedures , rather than duplicating code segments in multiple places.

Functions returning values after processing are called Functions and functions returning no value are called Sub Procedures. The Sub procedure performs a task and then returns control to the calling code, but it does not return a value to the calling code.

function sample

```
int add(int a, int b)
{
        int result = 0;
        result = a + b;
        return result;          //here function return the result
}
```

Sub Procedure sample

```
Private Sub add(ByVal a As Integer, ByVal b As Integer)
   Dim result As Integer
   result = a + b
   MsgBox("Result is  " & result)          //here Sub do not return the result
End Sub
```

4.
(a) Write a sample code to connect the database and display the query in Asp.net. – 8

Answer:

Introduction

This article explains how to connect the SQL database into ASP.net using C# language and how to insert and view the data, using Visual Studio 2015 Update 3.

Requirements

- Visual Studio 2015 Update 3
- ASP.NET 4.5.2
- SQL Server

If you want to connect to the SQL database into ASP.NET, using C#, it should follow the steps given below.

Step s….

5. (a) What is AJAX? Where are it used? – 5

Answer:

Ajax is not a programming language or a tool, but a concept. Ajax is a client-side script that communicates to and from a server/database without the need for a postback or a complete page refresh. The best definition I've read for Ajax is "the method of exchanging data with a server, and updating parts of a web page – without reloading the entire page." Ajax itself is mostly a generic term for various JavaScript techniques used to connect to a web server dynamically without necessarily loading multiple pages. In a more narrowly-defined sense, it refers to the use of XmlHttpRequest objects to interact with a web server dynamically via JavaScript.

Benefits of Ajax

There are 4 main benefits of using Ajax in web applications:

1. **Callbacks:** Ajax is used to perform a callback, making a quick round trip to and from the server to retrieve and/or save data without posting the entire page back to the server. By not performing a full postback and sending all form data to the server, network utilization is minimized and quicker operations occur. In sites and locations with restricted bandwidth, this can greatly improve network performance. Most of the time, the data being sent to and from the server is minimal. By using callbacks, the server is not required to process all form elements. By sending only the necessary data, there is limited processing on the server. There is no need to process all form elements, process the ViewState, send images back to the client, or send a full page back to the client.

2. **Making Asynchronous Calls**: Ajax allows you to make asynchronous calls to a web server. This allows the client browser to avoid waiting for all data to arrive before allowing the user to act once more.

3. **User-Friendly:** Because a page postback is being eliminated, Ajax enabled applications will always be more responsive, faster and more user-friendly.

4. **Increased Speed:** The main purpose of Ajax is to improve the speed, performance and usability of a web application. A great example of Ajax is the movie rating feature on Netflix. The user rates a movie and their personal rating for that movie will be saved to their database without waiting for the page to refresh or reload. These movie ratings are being saved to their database without posting the entire page back to the server.

What is an example of AJAX?
AJAX Examples

- Simple Examples. A simple AJAX example. Create a simple XMLHttpRequest, and retrieve data from a TXT file. ...
- Request Header Information. Retrieve all header information of a resource (file) ...
- Request XML Files. Load an XML file with AJAX. ...
- Retrieve Server Data with PHP and ASP. Retrieve the content of a PHP file.

(b) How do you add website in ASp.net? – 5

To add a page to the Web application

1. Close the Default. aspx page. ...
2. In Solution Explorer, right-click the Web application name (in this tutorial the application name is BasicWebSite), and then click Add -> New Item. ...
3. Select the Visual C# -> Web templates group on the left. ...
4. Click Add to add the web page to your project.

6. (a) Define XML schema with examples. – 5

Answer:
XML schema is a language which is used for expressing constraint about XML documents. There are so many schema languages which are used now a days for example **Relax- NG and XSD** (XML schema definition). An XML schema is used to define the structure of an XML document.

To create an XML schema

Open an XML file in Visual Studio. On the menu bar, **choose XML > Create Schema**. An XML Schema document is created and opened for each namespace found in the XML file. ... The schemas can be saved to disk, added to your project, or discarded.
XML Schemas Support Data Types

- It is easier to describe allowable document content.
- It is easier to validate the correctness of data.
- It is easier to define data facets (restrictions on data)
- It is easier to define data patterns (data formats)
- It is easier to convert data between different data types.

(b) Write XML syntax rules, validator. – 5

Answer:
You must follow these rules when you create XML syntax:

- All XML elements must have a closing tag.
- XML tags are case sensitive.
- All XML elements must be properly nested.
- All XML documents must have a root element.

- Attribute values must always be quoted.

XML Documents Must Have a Root Element
XML documents must contain one **root** element that is the **parent** of all other elements:

```
<root>
  <child>
    <subchild>.....</subchild>
  </child>
</root>
```

In this example **<note>** is the root element:

```
<?xml version="1.0" encoding="UTF-8"?>
<note>
  <to>Tove</to>
  <from>Jani</from>
  <heading>Reminder</heading>
  <body>Don't forget me this weekend!</body>
</note>
```

The XML Prolog
This line is called the XML **prolog**:

```
<?xml version="1.0" encoding="UTF-8"?>
```

- ✓ The XML prolog is optional. If it exists, it must come first in the document.
- ✓ XML documents can contain international characters, like Norwegian øæå or French êèé.
- ✓ To avoid errors, you should specify the encoding used, or save your XML files as UTF-8.
- ✓ UTF-8 is the default character encoding for XML documents.
- ✓ Character encoding can be studied in our Character Set Tutorial.
- ✓ UTF-8 is also the default encoding for HTML5, CSS, JavaScript, PHP, and SQL.

Introduction to XML Validation

XML validation is a process done to check for a syntax error in an XML document to ensure that the document is well-written with the standard rules using either DTD or schema. A complete XML file is considered to be valid XML document unless it has correct syntax and constraints defined in it. To check for the validation document type definitions follow two schemas and it is checked for well-doing so no need to consider the data and structure of the XML document. For instance, if the data is the location and the first name, validation works if the respective values are provided without non-empty. Validation is preferred to make sure that XML file is sent between a client and the server which does the service of web service that is acknowledged correctly.

How does Validation work in XML?

- Validation Process is dealt with XML Parser in two different ways.
- One is Well-defined XML Document and another one is Valid XML Document.
- Validation is optional but when it is about data integrity it is highly recommended.
- In most of the cases, validation of XML is done through schemas, not by DTD.
- The XML parser plays the main role in Validation has two categories like validating and non-validating.
- Former validate against DTD and later doesn't validate XML with respective to DTD.

7. (a) Explain Database and data manipulation statement in SQL with example. -8

Answer:

What is data manipulation in database?

Data manipulation refers to **the process of adjusting data to make it organised and easier to read**. Data manipulation language, or DML, is a programming language that adjusts data by inserting, deleting and modifying data in a database such as to cleanse or map the data.

How manipulating data in SQL explain with example?

In SQL, the data manipulation language comprises the SQL-data change **statements**,which modify stored data but not the schema or database objects.

...

In the case of SQL, these verbs are:

1. SELECT … FROM … WHERE … (strictly speaking DQL)
2. SELECT … INTO …
3. INSERT INTO … VALUES …
4. UPDATE … SET … WHERE …
5. DELETE FROM … WHERE …

- ✓ *SELECT* – to query data in the database
- ✓ *INSERT* – to insert data into a table
- ✓ *UPDATE* – to update data in a table
- ✓ *DELETE* – to delete data from a table

(b) How to connect server side script to database? – 4

Answer:

There are three types of methods in PHP to connect MySQL database through backend: **MySQL**. **MySQLi**.

...

Create MySQL Database at the Localhost

1. Create Database.
2. Create a Folder in htdocs.
3. Create Database Connection File In PHP.
4. Create new php file to check your database connection.
5. Run the server.

```
<?php
$servername = "localhost";
$username = "root";
$password = "";

// Create connection
//$conn = new mysqli($servername, $username, $password);
$xyz = new mysqli($servername, $username, $password);

// Check connection
if ($xyz->connect_error)
{
  die("Connection failed: " . $xyz->connect_error);
}
echo "MySql database Connected successfully";
?>
```

8. Write short notes on: Any TWO 2 x 5 = 10

(a) SOAP

What is soap full form?

SOAP (**Simple Object Access Protocol**) is a standards-based web services access protocol that has been around for a long time.

What do you mean by soap?

1a : **a cleansing and emulsifying agent made** usually by action of alkali on fat or fatty acids and consisting essentially of sodium or potassium salts of such acids. b : a salt of a fatty acid and a metal. 2 : soap opera.

Why is soap important?

Many types of bacteria and viruses, including the new coronavirus (COVID-19), can live on your hands and enter your body when you touch your eyes, nose or mouth, or the food you eat. Washing your hands regularly with soap and water is one of the **most effective ways to remove these germs** and avoid getting sick.

(b) SQL

What is SQL?

SQL stands **for Structured Query Language**. A query language is a kind of programming language that's designed to facilitate retrieving specific information from databases, and that's exactly what SQL does. To put it simply, SQL is the language of databases

What is the importance of query in database?

Queries can perform many different functions in a database. Their most common function is **to retrieve specific data from the tables**. The data you want to see is usually spread across several tables, and queries allow you to view it in a single datasheet.

What is the most important thing in SQL?

SELECT is by far the most important statement, as well as the most frequently used statement in SQL. It's used to retrieve data from a datable. You can show everything from a table, a column or just some records based on the conditions.

(c) Transaction

What is called as transaction?

A transaction is **any kind of action involved in conducting business**, or an interaction between people. ... An important business deal can be called a transaction, particularly the buying or selling of goods, but you can call any exchange with another person a transaction.

Questions Answers 2074

- Why internet technology is becoming popular now a days? Differentiate between client side and server side scripting. 4 + 4

Why has the internet become popular?

internet is very popular because it has so much of dense networks containing a huge variety of information and have a system that you can access to that piece of information by your computer thus connecting to that a web like networks.

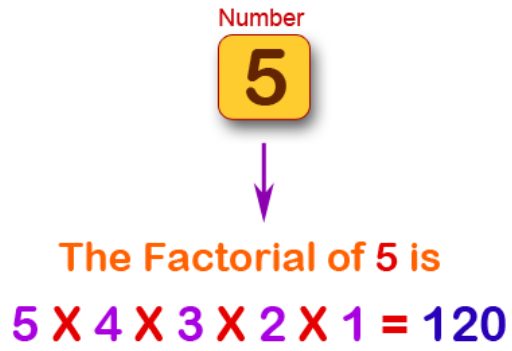
Why is technology important now a days?

Technology is inevitable in our everyday lives. ... The development of new technologies helps to save lives; it improves work and makes the world better. In reality, technology has played a very important role in how we live in the world today and how we communicate in the atmosphere with everything around us.

What is java script ? Write the java script to calculate the factorial of a number using function. 2 + 6

What is JavaScript and why it is used?
JavaScript is a text-based programming language used both on the client-side and server-side that allows you to make web pages interactive. Where HTML and CSS are languages that give structure and style to web pages, JavaScript gives web pages interactive elements that engage a user.

Write a JavaScript program to calculate the factorial of a number.
In mathematics, the factorial of a non-negative integer n, denoted by n!, is the product of all positive integers less than or equal to n. For example, 5! = 5 x 4 x 3 x 2 x 1 = 120



Sample Solution:-
HTML Code:

```
<!DOCTYPE html>
<html>
<head>
<meta charset=utf-8 />
<title>Calculate the factorial of a number.</title>
</head>
<body>
</body>
</html>
```

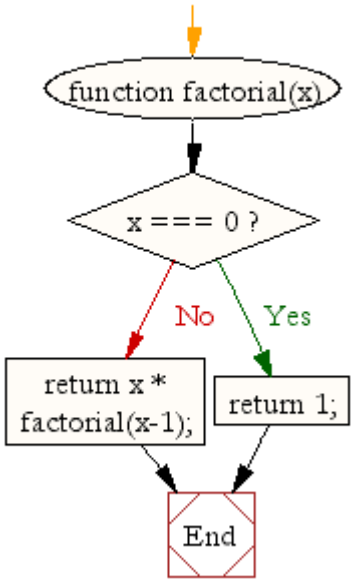
JavaScript Code:

```
function factorial(x)
{

if (x === 0)
{
return 1;
}
return x * factorial(x-1);
```

```
}
console.log(factorial(5));
```

Flowchart:



What are the different types of events in java script? Write a java script to explain about on click and on mouse out event. 3 + 5

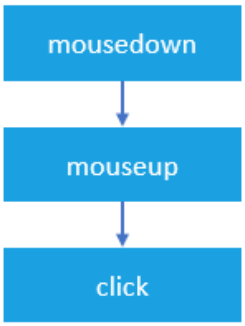
Answer:
JavaScript Event Types
User Interface events. These occur as the result of any interaction with the browser window rather than the HTML page. ...

- Focus and blur events. ...
- Mouse events. ...
- Keyboard events. ...
- Form events. ...
- Mutation events and observers. ...
- HTML5 events. ...
- CSS events

How do you use the mouse events in JavaScript explain?
If you depress the mouse button on an element and move your mouse off the element, and then release the mouse button. The only mousedown event fires on the element. Likewise, if you depress the mouse button, and move the mouse over the element, and release the mouse button, the only mouseup event fires on the element.

Introduction to JavaScript mouse events
Mouse events fire when you use the mouse to interact with the elements on the page. DOM Level 3 events define nine mouse events.
mousedown, mouseup, and click

When you click an element, there are no less than three mouse events fire in the following sequence:
The mousedown fires when you depress the mouse button on the element.
The mouseup fires when you release the mouse button on the element.
The click fires when one mousedown and one mouseup detected on the element.

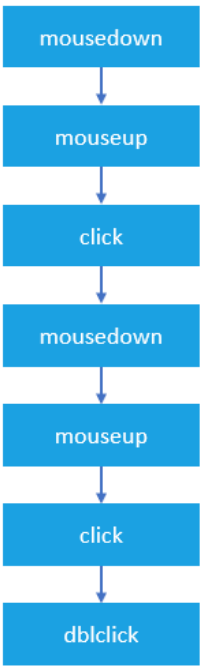


If you depress the mouse button on an element and move your mouse off the element, and then release the mouse button. The only mousedown event fires on the element.
Likewise, if you depress the mouse button, and move the mouse over the element, and release the mouse button, the only mouseup event fires on the element.
In both cases, the click event never fires.

dblclick

In practice, you rarely use the dblclick event. The dblclick event fires when you double click over an element.
It takes two click events to cause a dblclick event to fire. The dblclick event has four events fired in the following order:

mousedown
mouseup
click
mousedown
mouseup
click
dblclick



As you can see, the click events always take place before the dblclick event. If you register both click and dblclick event handlers on the same element, you will not know exactly what user actually has clicked or double-clicked the element.

mousemove

The mousemove event fires repeatedly when you move the mouse cursor around an element. Even when you move the mouse one pixel, the mousemove event still fires. It will cause the page slow, therefore, you only register mousemove event handler only when you need it and immediately remove the event handler as soon as it is no longer used, like this:

```
element.onmousemove = mouseMoveEventHandler;
// ...
//  later, no longer use
element.onmousemove = null;
```

Code language: JavaScript (javascript)

mouseover / mouseout

The mouseover fires when the mouse cursor is outside of the element and then move to inside the boundaries of the element.

The mouseout fires when the mouse cursor is over an element and then move another element.

mouseenter / mouseleave

The mouseenter fires when the mouse cursor is outside of an element and then moves to inside the boundaries of the element.

The mouseleave fires when the mouse cursor is over an element and then moves to the outside of the element's boundaries.

Both mouseenter and mouseleave does not bubble and does not fire when the mouse cursor moves over descendant elements.

Registering mouse event handlers

To register a mouse event, you use these steps:

First, select the element by using querySelector() or getElementById() method.

Then, register the mouse event using the addEventListener() method.

For example, suppose that you have the following button:

```
<button id="btn">Click Me!</button>
```

Code language: HTML, XML (xml)

To register a mouse click event handler, you use the following code:

```
let btn = document.querySelector('#btn');

btn.addEventListener('click',(event) => {
console.log('clicked');
});
```

Code language: JavaScript (javascript)

or you can assign a mouse event handler to the element's property:

```
let btn = document.querySelector('#btn');

btn.onclick = (event) => {
console.log('clicked');
};
```

Code language: JavaScript (javascript)

In legacy systems, you may find that the event handler is assigned in the HTML attribute of the element:

```
<button id="btn" onclick="console.log('clicked')">Click Me!</button>
```

Code language: HTML, XML (xml)
It's a good practice to always use the addEventListener() to register a mouse event handler.
Detecting mouse buttons
The event object passed to the mouse event handler has a property called button that indicates which mouse button was pressed on the mouse to trigger the event.
The mouse button is represented by a number:
0: the main mouse button pressed, usually the left button.
1: the auxiliary button pressed, usually the middle button or the wheel button.
2: the secondary button pressed, usually the right button.
3: the fourth button pressed, usually the *Browser Back* button.
4: the fifth button pressed, usually the *Browser Forward* button.

See the following example:

```
<!DOCTYPE html>
<html>
<head>
<title>JS Mouse Events - Button Demo</title>
</head>
<body>
<button id="btn">Click me with any mouse button: left, right, middle, ...</button>
<p id="message"></p>
<script>
let btn = document.querySelector('#btn');

// disable context menu when right-mouse clicked
btn.addEventListener('contextmenu', (e) => {
e.preventDefault();
});

// show the mouse event message
btn.addEventListener('mouseup', (e) => {
let msg = document.querySelector('#message');
switch (e.button) {
case 0:
msg.textContent = 'Left mouse button clicked.';
break;
case 1:
msg.textContent = 'Middle mouse button clicked.';
break;
case 2:
msg.textContent = 'Right mouse button clicked.';
```

```
break;
default:
msg.textContent = `Unknown mouse button code: ${event.button}`;
}
});
</script>
</body>
</html>
```

Code language: HTML, XML (xml)

In this example, when you click the button with your mouse (left-click, right-click, and middle-click), it shows a corresponding message on the element.

Click me with any mouse button: left, right, middle, …

Modifier keys

When you click an element, you may press one or more modifier keys: Shift, Ctrl, Alt, and Meta.

Note the Meta key is the Windows key on Windows keyboards and the Command key on Apple keyboard.

To detect if these modifier keys have been pressed, you can use the event object passed to the mouse event handler.

The event object has four Boolean properties, where each is set to true if the key is being held down or false if the key is not pressed.

See the following example:

```
<!DOCTYPE html>
<html>
<head>
<title>JS Modifier Keys Demo</title>
</head>
<body>
<button id="btnKeys">Click me with Alt, Shift, Ctrl pressed</button>
<p id="messageKeys"></p>

<script>
let btnKeys = document.querySelector('#btnKeys');

btnKeys.addEventListener('click', (e) => {
let keys = [];

if (e.shiftKey) keys.push('shift');
if (e.ctrlKey) keys.push('ctrl');
if (e.altKey) keys.push('alt');
if (e.metaKey) keys.push('meta');

let msg = document.querySelector('#messageKeys');
msg.textContent = `Keys: ${keys.join('+')}`;
```

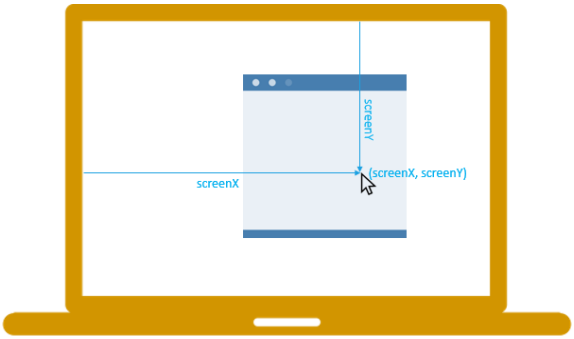
```
});
</script>
</body>
</html>
```

Code language: HTML, XML (xml)

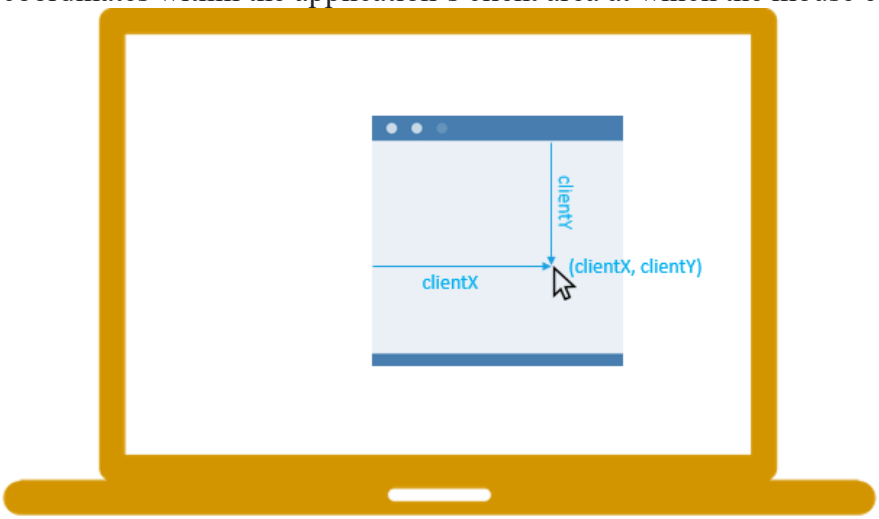
Click me with Alt, Shift, Ctrl pressed

Getting Screen Coordinates

The screenX and screenY properties of the event passed to the mouse event handler return the screen coordinates of the location of the mouse in relation to the entire screen.



On the other hand, the clientX and clientY properties provide the horizontal and vertical coordinates within the application's client area at which the mouse event occurred:



See the following demo:

```
<!DOCTYPE html>
<html>
<head>
<title>JS Mouse Location Demo</title>
<style>
#track {
background-color: goldenrod;
height: 200px;
width: 400px;
}
</style>
</head>
<body>
<p>Move your mouse to see its location.</p>
<div id="track"></div>
<p id="log"></p>
```

```
<script>
let track = document.querySelector('#track');
track.addEventListener('mousemove', (e) => {
let log = document.querySelector('#log');
log.innerText = `
Screen X/Y: (${e.screenX}, ${e.screenY})
Client X/Y: (${e.clientX}, ${e.clientY})`
});
</script>
</body>
</html>
```

Code language: HTML, XML (xml)
Move your mouse to see its location.

What is form validation? Explain event handling. 3 + 5
What is form validation?

Form validation is **a "technical process where a web-form checks if the information provided by a user is correct**." The form will either alert the user that they messed up and need to fix something to proceed, or the form will be validated and the user will be able to continue with their registration process.

What is an event handling functions?
Event handlers can be used to handle and verify user input, user actions, and browser actions: Things that should be done every time a page loads. Things that should be done when the page is closed. Action that should be performed when a user clicks a button. Events are signals fired inside the browser window that notify of changes in the browser or operating system environment. Programmers can create *event handler* code that will run when an event fires, allowing web pages to respond appropriately to change.

What is inheritance ? How object oriented programming is done with server side scripting. 2 + 6

What is Object Oriented Programming?
Object Oriented programming (OOP) is a programming paradigm that relies on the concept of **classes** and **objects**. It is used to structure a software program into simple, reusable pieces of code blueprints (usually called classes), which are used to create individual instances of objects. There are many object-oriented programming languages including JavaScript, C++, Java, and Python.

A **class** is an abstract blueprint used to create more specific, concrete objects. Classes often represent broad categories, like Car or Dog that share **attributes**. These classes define what attributes an instance of this type will have, like color, but not the value of those attributes for a specific object.

Classes can also contain functions, called **methods** available only to objects of that type. These functions are defined within the class and perform some action helpful to that specific type of object.

For example, our Car class may have a method repaint that changes the color attribute of our car. This function is only helpful to objects of type Car, so we declare it within the Car class thus making it a method.

Class templates are used as a blueprint to create individual **objects**. These represent specific examples of the abstract class, like myCar or goldenRetriever. Each object can have unique values to the properties defined in the class.

For example, say we created a class, Car, to contain all the properties a car must have, color, brand, and model. We then create an instance of a Car type object, myCar to represent my specific car.

We could then set the value of the properties defined in the class to describe my car, without affecting other objects or the class template.

We can then reuse this class to represent any number of cars.

Class blueprint being used to create two Car type objects, myCar and helensCar

Benefits of OOP

- OOP models complex things as reproducible, simple structures
- Reusable, OOP objects can be used across programs
- Allows for class-specific behavior through polymorphism
- Easier to debug, classes often contain all applicable information to them
- Secure, protects information through encapsulation

What are globalization and accessibility? Explain different server control methods. 4 + 4

What the globalization means?

Globalization means **the speedup of movements and exchanges** (of human beings, goods, and services, capital, technologies or cultural practices) all over the planet. One of the effects of globalization is that it promotes and increases interactions between different regions and populations around the globe.

What do you mean by accessibility?

Accessibility means that **people can do what they need to do in a similar amount of time and effort as someone** that does not have a disability. It means that people are empowered, can be independent, and will not be frustrated by something that is poorly designed or implemented.

Server Controls

Server Controls are **the tags that are understood by the server**. There are basically three types of server controls. ASP.NET provides a way to work with HTML Server controls on the server side; programming with a set of controls collectively is called HTML Controls.

What are the types of server control?

There are basically three types of server controls.

- HTML Server Controls - Traditional HTML tags.
- Web Server Controls - New ASP. NET tags.
- Validation Server Controls - For input validation.

Why database security is needed? Explain view & trigger. 3 + 5

Why do we need database security explain briefly?

Safeguarding the data your company collects and manages is of utmost importance. Database security **can guard against a compromise of your database**, which can lead to financial loss, reputation damage, consumer confidence disintegration, brand erosion, and non-compliance of government and industry regulation.

What is database security?

Database security encompasses a range of security controls designed to protect the Database Management System (DBMS). The types of database security measures your business should use include protecting the underlying infrastructure that houses the database such as the network and servers), securely configuring the DBMS, and the access to the data itself.

Database security controls

Database security encompasses multiple controls, including system hardening, access, DBMS configuration, and security monitoring. These different security controls help to manage the circumventing of security protocols.

System hardening and monitoring

The underlying architecture provides additional access to the DBMS. It is vital that all systems are patched consistently, hardened using known security configuration standards, and monitored for access, including insider threats.

DBMS configuration

It is critical that the DBMS be properly configured and hardened to take advantage of security features and limit privileged access that may cause a misconfiguration of expected security settings. Monitoring the DBMS configuration and ensuring proper change control processes helps ensure that the configuration stays consistent.

Authentication

Database security measures include authentication, the process of verifying if a user's credentials match those stored in your database, and permitting only authenticated users access to your data, networks, and database platform.

Access

A primary outcome of database security is the effective limitation of access to your data. Access controls authenticate legitimate users and applications, limiting what they can access in your database. Access includes designing and granting appropriate user attributes and roles and limiting administrative privileges.

Database auditing

Monitoring (or auditing) actions as part of a database security protocol delivers centralized oversight of your database. Auditing helps to detect, deter, and reduce the overall impact of unauthorized access to your DBMS.

Backups

A data backup, as part of your database security protocol, makes a copy of your data and stores it on a separate system. This backup allows you to recover lost data that may result from hardware failures, data corruption, theft, hacking, or natural disasters.

Encryption

Database security can include the secure management of encryption keys, protection of the encryption system, management of a secure, off-site encryption backup, and access restriction protocols.

Application security
Database and application security framework measures can help protect against common known attacker exploits that can circumvent access controls, including SQL injection.

Why is database security important?
Safeguarding the data your company collects and manages is of utmost importance. Database security can guard against a compromise of your database, which can lead to financial loss, reputation damage, consumer confidence disintegration, brand erosion, and non-compliance of government and industry regulation.
Database security safeguards defend against a myriad of security threats and can help protect your enterprise from:

- Deployment failure
- Excessive privileges
- Privilege abuse
- Platform vulnerabilities
- Unmanaged sensitive data
- Backup data exposure
- Weak authentication
- Database injection attacks

What is Database & DBMS ? Explain database design ? 2 + 6
What is a Database Management System (DBMS)?
A database management system is a software used to perform different operations, like ***addition, access, updating, and deletion of the data,*** like adding your name in the database for an online retail store as a customer. A database management system acts as the backbone of a database and makes using a database a cakewalk as it makes access and management of data a lot easier.
One thing we need to understand is the difference between ***Database*** and ***Database Management System.***

What is Database (DB)?
So what is a Database? Is it a random collection of stuff all squeezed in together?
No, ***Database (DB)*** are organized, they have a structure, and all the data they store it fits into that structure. More specifically, a database is an electronic system that allows data to be ***stored***, easily ***accessed, manipulated and updated***.

What is Data?
We all are familiar with ***Social*** websites like ***Facebook & Instagram, E-Commerce*** websites like ***Amazon & SnapDeal, Banking*** websites like ***Barclays & HDFC*** or any ***Education*** website like ***ToolsQA,*** all these websites have a humongous amount of data stored in it.

What are the rules for XML syntax? Explain about XSLT. 2 + 2 + 4
What is XML explain its syntax?
XML (Extensible Markup Language) is a markup language **like HTML for storage or transmission of data**. XML is widely used in web services to transport data over the network. XML has no predefined tags, unlike HTML. XML is very easy to parse and generate.

Which are the XML syntax rules?

- **XML Syntax Rules**
- XML Documents Must Have a Root Element. ...
- The XML Prolog. ...
- All XML Elements Must Have a Closing Tag. ...
- XML Tags are Case Sensitive. ...
- XML Elements Must be Properly Nested. ...
- XML Attribute Values Must Always be Quoted. ...
- Entity References. ...
- Comments in XML.

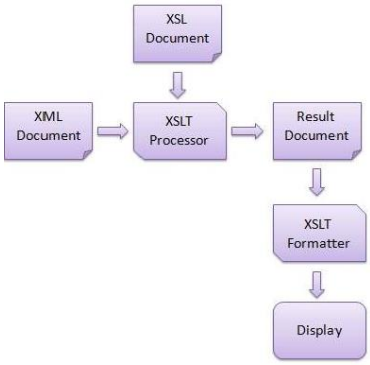
What is XSLT?

Before XSLT, first we should learn about XSL. XSL stands for EXtensible Stylesheet Language. It is a styling language for XML just like CSS is a styling language for HTML.
XSLT stands for XSL Transformation. It is used to transform XML documents into other formats (like transforming XML into HTML).

How XSLT Works?

The XSLT stylesheet is written in XML format. It is used to define the transformation rules to be applied on the target XML document. The XSLT processor takes the XSLT stylesheet and applies the transformation rules on the target XML document and then it generates a formatted document in the form of XML, HTML, or text format. At the end it is used by XSLT formatter to generate the actual output and displayed on the end-user.

Image representation:



Advantage of XSLT

A list of advantages of using XSLT:
XSLT provides an easy way to merge XML data into presentation because it applies user defined transformations to an XML document and the output can be HTML, XML, or any other structured document.

XSLT provides Xpath to locate elements/attribute within an XML document. So it is more convenient way to traverse an XML document rather than a traditional way, by using scripting language.

XSLT is template based. So it is more resilient to changes in documents than low level DOM and SAX.
By using XML and XSLT, the application UI script will look clean and will be easier to maintain.
XSLT templates are based on XPath pattern which is very powerful in terms of performance to process the XML document.

XSLT can be used as a validation language as it uses tree-pattern-matching approach.
You can change the output simply modifying the transformations in XSL files.

XSLT Syntax

Let's take an example to take a sample XML file and transform it into a well formatted HTML document.

See this example:

Create an XML file named employee.xml, having the following code:

Employee.xml

```
<?xml version = "1.0"?>
<class>
  <employee id = "001">
    <firstname>Aryan</firstname>
    <lastname>Gupta</lastname>
    <nickname>Raju</nickname>
    <salary>30000</salary>
  </employee>
  <employee id = "024">
    <firstname>Sara</firstname>
    <lastname>Khan</lastname>
    <nickname>Zoya</nickname>
    <salary>25000</salary>
  </employee>
  <employee id = "056">
    <firstname>Peter</firstname>
    <lastname>Symon</lastname>
    <nickname>John</nickname>
    <salary>10000</salary>
  </employee>
</class>
```

Define an XSLT stylesheet document for the above XML document. You should follow the criteria give below:

Page should have a title employee.
Page should have a table of employee's details.
Columns should have following headers: id, First Name, Last Name, Nick Name, Salary
Table must contain details of the employees accordingly.

Step1: Create XSLT document

Create the XSLT document which satisfies the above requirements. Name it as employee.xsl and save it in the same location of employee.xml.

Employee.xsl

```
<?xml version = "1.0" encoding = "UTF-8"?>
<!-- xsl stylesheet declaration with xsl namespace:
Namespace tells the xlst processor about which
element is to be processed and which is used for output purpose only
-->
<xsl:stylesheet version = "1.0"
xmlns:xsl = "http://www.w3.org/1999/XSL/Transform">
<!-- xsl template declaration:
template tells the xlst processor about the section of xml
document which is to be formatted. It takes an XPath expression.
In our case, it is matching document root element and will
```

```
tell processor to process the entire document with this template.
-->
  <xsl:template match = "/">
    <!-- HTML tags
      Used for formatting purpose. Processor will skip them and browser
        will simply render them.
    -->
    <html>
      <body>
        <h2>Employee</h2>
        <table border = "1">
          <tr bgcolor = "#9acd32">
            <th>ID</th>
            <th>First Name</th>
            <th>Last Name</th>
            <th>Nick Name</th>
            <th>Salary</th>
          </tr>
          <!-- for-each processing instruction
          Looks for each element matching the XPath expression
          -->
          <xsl:for-each select="class/employee">
            <tr>
              <td>
                <!-- value-of processing instruction
                process the value of the element matching the XPath expression
                -->
                <xsl:value-of select = "@id"/>
              </td>
              <td><xsl:value-of select = "firstname"/></td>
              <td><xsl:value-of select = "lastname"/></td>
              <td><xsl:value-of select = "nickname"/></td>
              <td><xsl:value-of select = "salary"/></td>
            </tr>
          </xsl:for-each>
        </table>
      </body>
    </html>
  </xsl:template>
</xsl:stylesheet>
```

Step2: List the XSLT document to the XML document

Update employee.xml document with the following xml-stylesheet tag. Set href value to employee.xsl

```
<?xml version = "1.0"?>
<?xml-stylesheet type = "text/xsl" href = "employee.xsl"?>
<class>
...
</class>
```

Updated "employee.xml"

```
<?xml version = "1.0"?>
<?xml-stylesheet type = "text/xsl" href = "employee.xsl"?>
```

```
<class>
  <employee id = "001">
    <firstname>Aryan</firstname>
    <lastname>Gupta</lastname>
    <nickname>Raju</nickname>
    <salary>30000</salary>
  </employee>
  <employee id = "024">
    <firstname>Sara</firstname>
    <lastname>Khan</lastname>
    <nickname>Zoya</nickname>
    <salary>25000</salary>
  </employee>
  <employee id = "056">
    <firstname>Peter</firstname>
    <lastname>Symon</lastname>
    <nickname>John</nickname>
    <salary>10000</salary>
  </employee>
</class>
```

Step3: View the XML document in Internet Explorer

The output will look like this:

Output:

Employee

ID	First Name	Last Name	Nick Name	Salary
001	Aryan	Gupta	Raju	60000
024	Sara	Khan	Zoya	45000
056	Peter	Symon	John	20000

Exam Questions: 2075

How can you add Javascript in a web page?

Adding JavaScript to Your Web Pages

There are typically three ways to add JavaScript to a web page:

Embedding the JavaScript code between a pair of <script> and </script> tag.

Creating an external JavaScript file with the .js extension and then load it within the page through the src attribute of the <script> tag.

Placing the JavaScript code directly inside an HTML tag using the special tag attributes such as onclick, onmouseover, onkeypress, onload, etc.

Explain the concept of object oriented programming with Javascprit. Explain image and form objects in Javascript. 4 + 6

What is object-oriented programming in JavaScript?

Object-oriented programming allows developers to build applications with reusable and maintainable blocks of code, which leads to efficiency and simplified software design. With object-oriented JavaScript, you'll be able **to build classes to construct objects that encapsulate both data and functionality**.

```
<!DOCTYPE html>
<html>
<head>
<title>JavaScript Dynamic Images using src property</title>
<script>
var demoImages = new Array("ab.jpg", "d1.jpg",
"d3.jpg", "dollar-bill.jpg");

index_no = 0;

function next_frame(){
index_no++;
if(index_no < demoImage.length){
document.images["money"].src = demoImages[index_no];
}
else{
index_no = 0;
document.images["money"].src = demoImages[index_no];
}
}
function last_frame(){
index_no--;
if(index_no >= 0){
document.images["money"].src = demoImages[index_no];
}
else{
index_no = demoImages.length - 1;
document.images["money"].src = demoImages[index_no];
}
}
</script>
</head>
<body>
<img name="money" id="money" src="dollar-bill.jpg"
width="400" height="200">
<br>

<a href="JavaScript: next_frame()"> Next Frame </a> <br>
<a href="JavaScript: last_frame()">Previous Frame </a> <br>


</body>
</html>
```

What is form validation? Explain Event Handling in Javascript. 4 + 6

What is form validation?

Form validation is **a "technical process where a web-form checks if the information provided by a user is correct**." The form will either alert the user that they messed up and need to fix something to proceed, or the form will be validated and the user will be able to continue with their registration process.

What is JavaScript event handling?

The change in the state of an object is known as an Event. This process of reacting over the events is called Event Handling. ... Thus, **js handles the HTML events via Event Handlers**. For example, when a user clicks over the browser, add js code, which will execute the task to be performed on the event.

What is dotnet framework? Explain its application. 4 + 6
What is .Net Framework?
.Net Framework is a software development platform developed by Microsoft for building and running Windows applications. The .Net framework consists of developer tools, programming languages, and libraries to build desktop and web applications. It is also used to build websites, web services, and games.

The .Net framework was meant to create applications, which would run on the Windows Platform. The first version of the .Net framework was released in the year 2002. The version was called .Net framework 1.0. The Microsoft .Net framework has come a long way since then, and the current version is .Net Framework 4.7.2.

The Microsoft .Net framework can be used to create both - **Form-based** and **Web-based** applications. Web services can also be developed using the .Net framework.
The framework also supports various programming languages such as Visual Basic and C#. So developers can choose and select the language to develop the required application. In this chapter, you will learn some basics of the .Net framework.

What is .NET Framework application?
NET Framework is **a software development framework for building and running applications on Windows**. ... NET Framework is part of the . NET platform, a collection of technologies for building apps for Linux, macOS, Windows, iOS, Android, and more.

What do you understand by procedures ? Explain Sub-routines and functions in brief. 2 + 4 + 4

How do you explain procedures?
Here are some good rules to follow:
Write actions out in the order in which they happen. ...

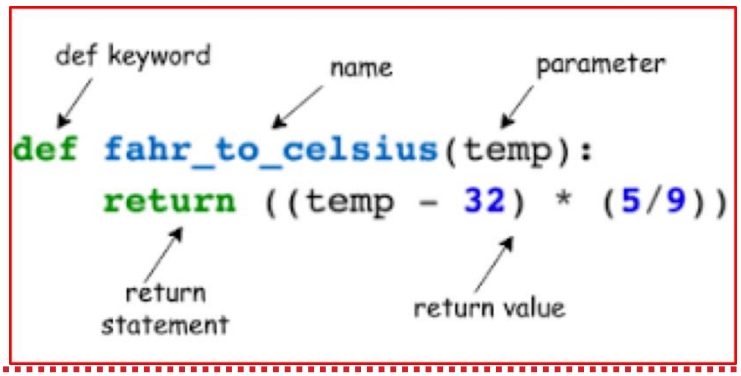
- Avoid too many words. ...
- Use the active voice. ...
- Use lists and bullets.
- Don't be too brief, or you may give up clarity.
- Explain your assumptions, and make sure your assumptions are valid.
- Use jargon and slang carefully.

Subroutines (Procedures and Functions)

Table of Contents

Fork me on GitHub

1 Subroutines - Procedures and Functions



Learn It: What are subroutines?

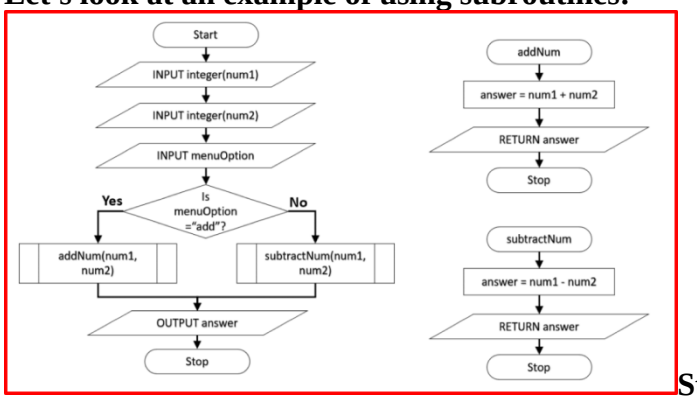
Subroutines - In computer programming, a subroutine is a sequence of program instructions
that performs a specific task, packaged as a unit. In different programming languages, a subroutine may be called a procedure, a function, a routine, a method, or a subprogram. The generic term 'callable unit' is sometimes used. (Source Wikipedia.org)
Parameters - Parameters or arguments are pieces of data that are passed into a subroutine in order for that subroutine to correctly function.
In this example: def displayTotal(total): 'total' is a parameter that will be passed to the 'displayTotal' subroutine. Multiple parameters can be managed using commas, for example: def displayTotals(total1, total2, total3, etc)

Subroutines are sequences of instructions that **perform a specific task**. It **may return** one or more values, *but does not have to*.

It may be easier to think of them as mini-programs **within** a large program.
Subroutines **consist** of modules of code that **perform different tasks**.

If these **tasks** are **repeated** throughout the program, they can be written as subroutines.
Each subroutine is given a unique name so that it can be called and executed quickly *throughout the program*, **without** having to **write the code again**.

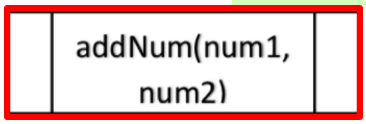
This reduces the size of the code, making
the program more logical and easier to read and maintain.
Let's look at an example of using subroutines:



Step 1
Here the main program (*shown in the flowchart on the left, beginning with the word "Start"*) asks the user to input two numbers.
It then asks them to **select an option** from a menu.

If they enter the word **"add"**, it will run a subroutine called addNum (*The smaller flowchart on the top right*), if they enter anything else it will run the subtractNum subroutine (*bottom right*).

Let's look at the subroutine box in the main flowchart first:



Step 2

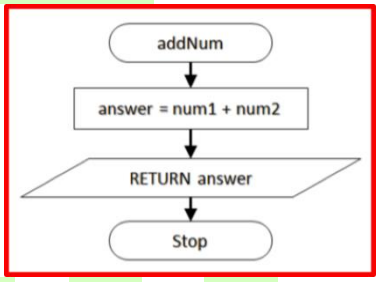
We can see the **name** of the subroutine has been **defined** along with the **variables**; num1 and num2.

This means that the **addNum subroutine** is going to use the values of num1 and num2 and so has been passed these **values**.

Sending variables to a **subroutine** is **known** as =passing *the variables*.

Step 3

We can now look at the addNum subroutine flowchart.



This subroutine will add together the num1 and num2 variables and store these in a **new variable** called answer.

It will then return the **variable answer back** to the main menu.
It will then **go back** to the main menu, *once the addNum subroutine* has been **completed**, where it will be able to display the answer variable.

If we had not returned the answer variable back to the main menu, it would not be able to display the answer in the output as the main program on its own has not defined the answer variable and it does not know what it is.

Learn It: Advantages of using Subroutines in programs

Advantages of using Subroutines

Breaking down or **decomposing** a complex programming task into **smaller sub-tasks** and **writing** each of these as subroutines, makes the **problem** easier to solve.
Subroutines can be used several times within a program.
It saves the programmer time as it **reduces the amount of code** that needs to be written or amended by allowing you to reuse code **without** having to write it again.
If you are working as part of a team you can divide a large program into smaller sections and allow individuals to simultaneously work on those sections.
It makes the code easier to read if you use sensible subroutine labels as the headings tell the reader what that section of code is doing.
By **reducing** the amount of **repeating tasks** you also **reduce the risk** of introducing errors in a program.
Easy to maintain as each subroutine can be tested separately.

Learn It: Passing Data within Programs

A parameter is a variable that is "passed" to a subroutine.

Using Parameters to pass data within Programs

A parameter is a variable used in a **subroutine** to refer to data that is **inputted** *into the subroutine*.

If a subroutine requires a value that has been used in **another part of the program**, then this variable **has** to be **"passed"** to the subroutine.

You can have more than one **parameter** passing into a **subroutine at a time**.

```
1  def addNum(num1, num2):
2      total = num1 + num2
3      print(total)
4
5
6  num1 = int(input("Please enter a number to be added: "))
7  num2 = int(input("Please enter another number to be added: "))
8  addNum(num1, num2)
```

```
Shell
>>> %Run addNum_Sub.py
  Please enter a number to be added: 10
  Please enter another number to be added: 20
  30
```

This example demonstrates how to use **more than one parameter** to be passed into our **subroutine**.

In this case num1 and num2 are **variables** inputted by the user in the main program.

Once the variables have been inputted, the addNum subroutine is **called** and the two parameters (*num1 and num2*) are passed to that subroutine so they can be used.

Learn It: Returning values from a Subroutine

A parameter is a variable that is "passed" to a subroutine.

Returning Single Values

We can **alter the previous code example** and **instead** of outputting the total as part of the subroutine, we can use the variable in the main program or even in another subroutine to pass that value back into the **main (calling) program** as shown below:

```
1  def addNum(num1, num2):
2      total = num1 + num2
3      return (total)
4
5
6  num1 = int(input("Please enter a number to be added: "))
7  num2 = int(input("Please enter another number to be added: "))
8  addNum(num1, num2)
9  print (addNum(num1, num2))
```

```
Shell
>>> %Run addNum_Sub.py
  Please enter a number to be added: 10
  Please enter another number to be added: 20
  30
```

Returning Multiple Values

We can **return** more than one value **back** to the main (calling) program, however you need to create something called a tuple.

A tuple is a short list that holds **values temporarily**.

The following pseudocode example, returns the numbers and the total back to the main program.

```
SUBROUTINE addition()
      num1 ← integer(USERINPUT)
      num2 ← integer(USERINPUT)
      total ← num1 + num2
      returningValues = (num1,num2,total)
      RETURN returningValues
ENDSUBROUTINE

name ← USERINPUT
num1,num2,total ← addition()
OUTPUT "Hello" + name
OUTPUT "You entered" + num1 + "and" + num2
OUTPUT "The total is"+ total
```

Here num1, num2 and total have been combined in the **subroutine** into a tuple called returningValue and this is retuned to the program using the line num1, num2, total ← addition().
This will only work as long as the data is presented in the same order as it appears in the tuple.
The **values** that are being returned are grouped in a tuple and then used in the main program, once they have been returned in the same order as the tuple.
How to connection with web server in PHP/MySql

```
<?php
$servername = "localhost";
$username = "root";
$password = "";

// Create connection
//$conn = new mysqli($servername, $username, $password);
$xyz = new mysqli($servername, $username, $password);

// Check connection
if ($xyz->connect_error)
{
die("Connection failed: " . $xyz->connect_error);
}
echo "MySql database Connected successfully";
?>
```

How to create database with PHP code.

```
<?php
$servername = "localhost";
$username = "root";
$password = "";

// Create connection
$conn = new mysqli($servername, $username, $password);
// Check connection
if ($conn->connect_error) {
die("Connection failed: " . $conn->connect_error);
}

// Create database
$sql = "CREATE DATABASE myDB";
```

```
if ($conn->query($sql) === TRUE) {
echo "Database created successfully";
} else {
echo "Error creating database: " . $conn->error;
}

$conn->close();
?>
```

Explain creating and connecting Database with server-side scripting with code examples. 5 + 5
Define 'XML' with its syntax. Explain 'XML' schema with example. 5 + 5
What is XML Schema with example?
XML schema is a language which is **used for expressing constraint about XML documents**. There are so many schema languages which are used now a days for example Relax- NG and XSD (XML schema definition). An XML schema is used to define the structure of an XML document.

Write Short Notes on (Any Two) 2 x 5 = 10
What is SQL explain?
Basically, SQL stands **for Structured Query Language** which is basically a language used by databases. This language allows to handle the information using tables and shows a language to query these tables and other objects related (views, functions, procedures, etc.).

What is jQuery is used for?
jQuery is a lightweight, "write less, do more", **JavaScript library**. The purpose of jQuery is to make it much easier to use JavaScript on your website. jQuery takes a lot of common tasks that require many lines of JavaScript code to accomplish, and wraps them into methods that you can call with a single line of code.

What is jQuery with an example?
jQuery uses CSS-style selectors to select parts, or elements, of an HTML page. ... In jQuery, the class and ID selectors are like those in CSS. Here's an example of a jQuery method that selects **all paragraph elements**, and adds a class of "selected" to them: <p>This is a paragraph selected by a jQuery method.

Interfaces and collection
What is collection interface?
A Collection represents a group of objects known as its elements. The Collection interface is **used to pass around collections of objects where maximum generality is desired**. For example, by convention all general-purpose collection implementations have a constructor that takes a Collection argument.

How many interfaces are in a collection?
The collection hierarchy consists of **six interfaces**, the core collection intefaces. Three of these interfaces, Set, List, and SortedSet are descendants of the Collection interface; they add further constraints on the contracts imposed by the methods in this interface, as well as adding new methods.

Best Of Luck