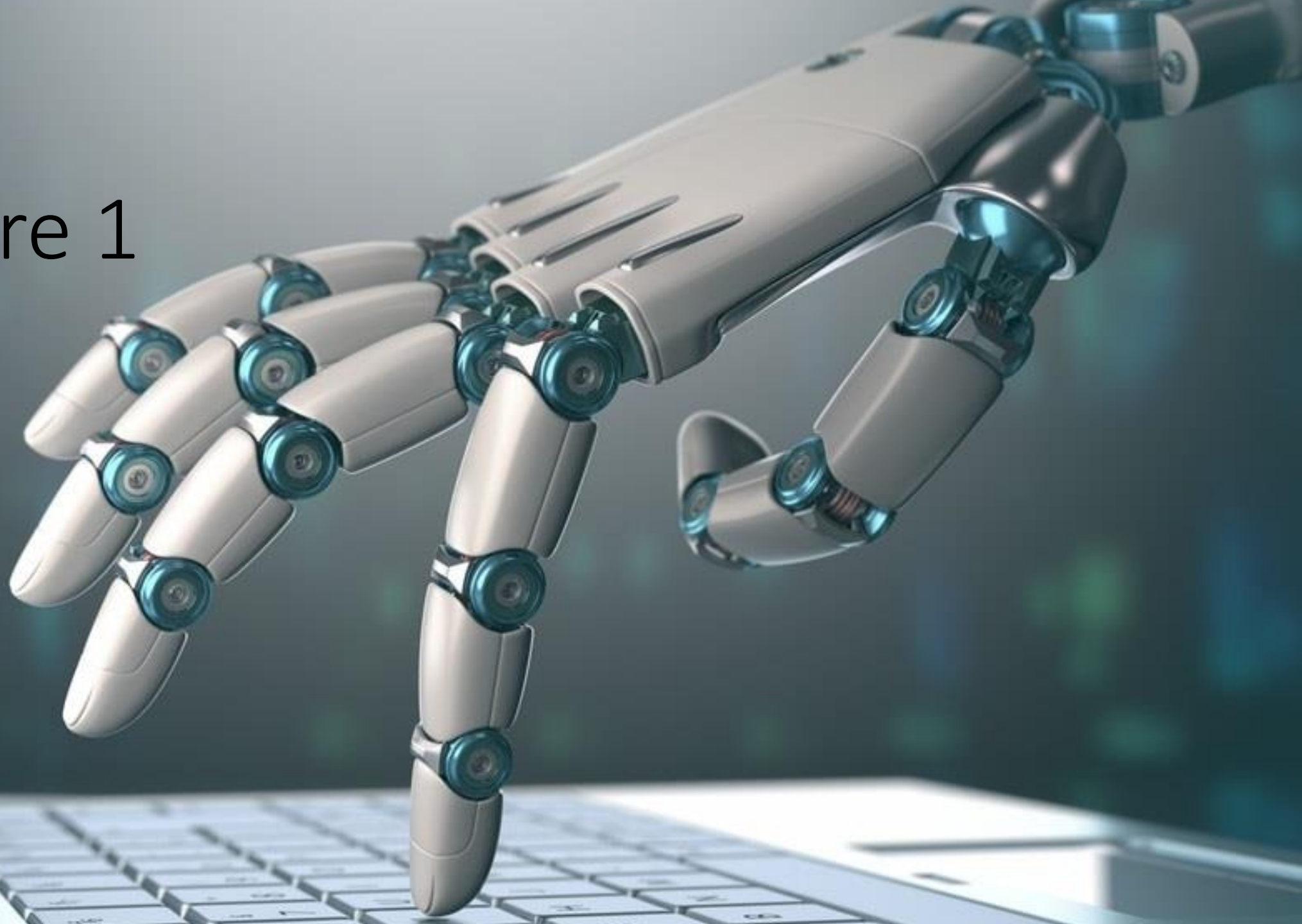


# Lecture 1





# Agenda

- **About Course**
- Automation reminder
- Automation types
- Basic concepts of web applications
- What is Website?
- DOM
- HTML, CSS, JS - Basics
- Why Java?
- Java variables
- Arithmetic operators
- Practice

# About the course

- Theory and Practice (12 lessons)
- To get **certificate** – **MUST** complete and lecturer must **accept FINAL PROJECT**
- Homework – 4 obligatory – 2 group tasks + optional
- Ask as many questions as you need
- Reference and links added additionally on git and after each lecture – reference part
- Hot seat – programming
- Group tasks

# Course Content

1. Web and Intro to Java
2. GIT - OOP - Pure Java
3. Flow Control – Pure Java
4. Loops and Arrays – Pure Java
5. Git
6. Asserts and Libraries
7. Selenium
8. Pom and Advance Locators
9. Architecture and Data Models
10. Cucumber
11. Selenide and Final Project Start
12. Demo of final project

# Why Course content is such

1. Build Strong programming skills
2. Get understanding of automation basics
3. Latest news in IT field – for automation
4. Latest new on technologies that currently used on the market
5. Boost your skills and have fun!
6. Have fun and make new friends from your future colleagues
7. Anybody can finish this – slow pace
8. Building own **BIG** portfolio project
9. Many stories from people who finished this course – have found their dream job in IT (Manual QA, Development, Automation QA)

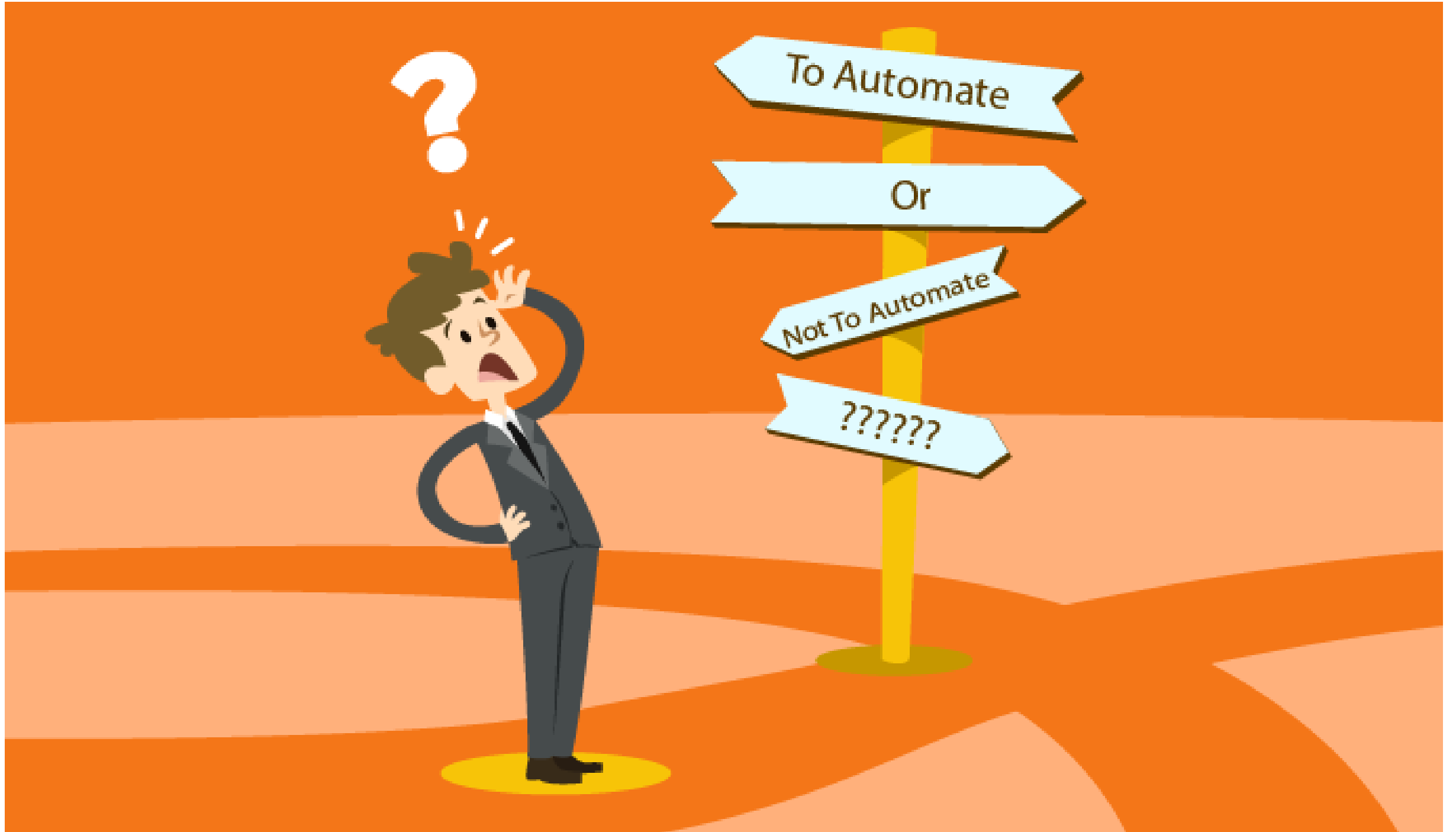
# Communication with Us

- **Telegram** chat
- **Email** – ask you lecturer
- **Homework** are sent to your lecturer as git links to git classes
- After the course – please leave a **feedback** – it is very **important** to get **your feeling** about the course

# Agenda

- About Course
- **Automation reminder**
- Automation types
- Basic concepts of web applications
- What is Website?
- DOM
- HTML, CSS, JS - Basics
- Why Java?
- Java variables
- Practice





# Test cases to automate

- Business critical paths – the features or user flows that if they fail, cause a considerable damage to the business.
- Tests that need to be run against every build/release of the application, such as smoke test, sanity test and regression test.
- Tests that need to run against multiple configurations — different OS & Browser combinations.
- Tests that execute the same workflow but use different data for its inputs for each test run.
- Tests that involve inputting large volumes of data, such as filling up very long forms.
- Tests that can be used for performance testing, like stress and load tests.
- Tests that take a long time to perform and may need to be run during breaks or overnight.
- Tests during which images must be captured to prove that the application behaved as expected, or to check that a multitude of web pages looks the same on multiple browsers.



# Test cases not to automate



NO

- Tests that you will only run only once. The only exception to this rule is that if you want to execute a test with a very large set of data, even if it's only once, then it makes sense to automate it.
- User experience tests for usability (tests that require a user to respond as to how easy the app is to use).
- Tests that need to be run ASAP. Usually, a new feature which is developed requires a quick feedback so testing it manually at first.
- Tests that require ad hoc/random testing – Exploratory Testing.
- Intermittent tests. Tests without predictable results cause more noise than value. To get the best value out of automation the tests must produce predictable and reliable results in order to produce pass and fail conditions.
- Tests that require visual confirmation, however, we can capture page images during automated testing and then have a manual check of the images.
- Test that cannot be 100% automated should not be automated at all, unless doing so will save a considerable amount of time.

# Agenda

- About Course
- Automation reminder
- **Automation types**
- Basic concepts of web applications
- What is Website?
- DOM
- HTML, CSS, JS - Basics
- Why Java?
- Java variables
- Practice

# Automation types

- Web testing
- Desktop app testing
- Mobile app testing
- Back-end testing
- Security/Penetration testing
- Visual testing
- Game testing
- Etc.

# Automation types breakdown

- Web testing – most commonly used
- Desktop app testing - too old/ rarely used
- Mobile app testing – commonly still web /time to time used
- Back-end testing – time to time used
- Security/Penetration testing – skill needed to test this stuff
- Visual testing – time to time used
- Game testing –rarely/currently tested only manually
- Etc.
- **Explanation:** *most commonly -> commonly -> time to time -> rarely -> tested only manually*

# Web testing

- Web testing
- Mobile Web testing
- Stress/Load/Performance testing
- Security testing
- Visual testing
- Usability testing

# Desktop App Testing

- App testing
- Stress/Load/Performance testing
- Security testing
- Visual testing
- Usability testing





# Mobile App testing

- Mobile App testing
- Stress/Load/Performance testing
- Security testing
- Visual testing
- Usability testing
- Game testing

# Back-End testing

- Rest testing
- Soap testing
- Stress/Load/Performance testing
- DB testing
- Security testing
- Etc.

# Security Testing

- Security testing
- Penetration testing
- Social Engineering
- Best lectures from Kevin Mitnick – one of the first hackers

# Visual testing

- Visual testing
- Pictures
- Buttons
- Font
- GUI
- Etc.

# Game Testing

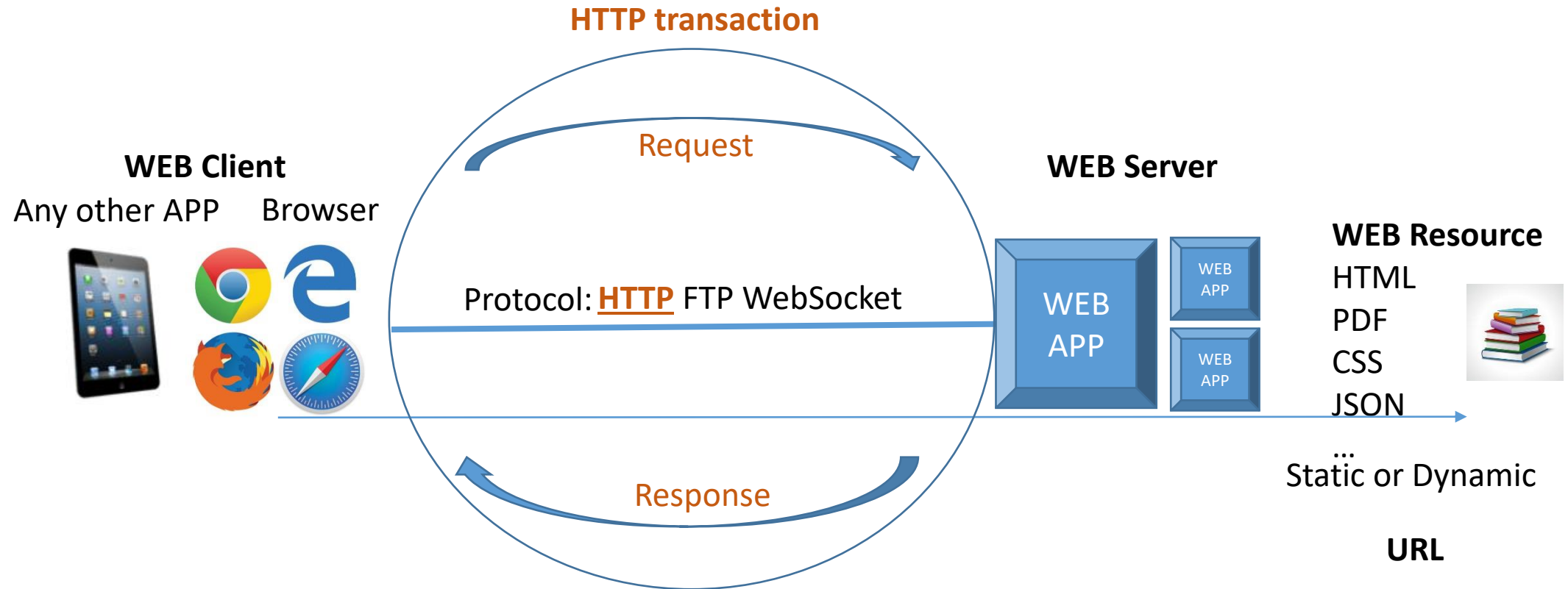
- Stress/Load/Performance testing
- Security testing
- Visual testing
- Usability testing
- Mobile device testing
- Game logic
- Game balance
- Game difficulty
- ...

# Agenda

- About Course
- Automation reminder
- Automation types
- **Basic concepts of web applications**
- What is Website?
- DOM
- HTML, CSS, JS - Basics
- Why Java?
- Java variables
- Practice

# Basic concepts of web applications

## How the www works



URL: **Protocol** **Domain->WEB server** **WEB Resource**  
*https://www.ss.com/msg/lv/transport/cars/dodge/charger/fmdfp.html*

# What is Browser?

## The Web Browser

A web browser (commonly referred to as a browser) is a software application for retrieving, presenting and traversing information resources on the World Wide Web.

An information resource is identified by a Uniform Resource Identifier (URI/URL) and may be a web page, image, video or other piece of content.

Popular web browsers include Firefox, Internet Explorer, Google Chrome, Opera, and Safari.

Quick Heal

AntiVirus





# Magnificent 5



Or is it so? What browsers are really important?



## Usage share of all browsers

Browser ◆	StatCounter <sup>[15]</sup> ◆ October 2021	NetMarketShare <sup>[16]</sup> ◆ October 2021	Wikimedia <sup>[17]</sup> ◆ October 2021
Chrome	64.67%	66.64%	52.5%
Safari	19.06%	13.92%	23.9%
Edge	4.10%	4.55%	3.0%
Firefox	3.66%	2.18%	4.4%
Samsung Internet	2.81%	3.04%	2.2%
Opera	2.36%	3.02%	1.0%
Others	3.34%	6.65%	13.0%

### Usage share of desktop browsers

Browser ↕	StatCounter <sup>[18]</sup> October 2021 ↕	NetMarketShare <sup>[19]</sup> October 2021 ↕	W3Counter <sup>[20]</sup> September 2021 ↕	Wikimedia <sup>[21]</sup> October 2021 ↕
Chrome	67.17%	72.96%	63.3%	58.0%
Safari	9.63%	2.72%	17.7%	9.3%
Edge	9.33%	12.61%	5.4%	7.8%
Firefox	7.87%	5.54%	5.8%	10.7%
Opera	2.89%	1.01%	1.3%	2.0%
<i>Others</i>	3.11%	5.56%	6.5%	12.2%

### Usage share of mobile browsers

Browser	StatCounter <sup>[22]</sup> October 2021	NetMarketShare <sup>[23]</sup> October 2021	Wikimedia <sup>[24]</sup> October 2021
Chrome	63.57%	63.07%	48.1%
Safari	25.61%	19.01%	32.2%
Samsung Internet	5.17%	5.00%	3.5%
Opera	2.02%	4.47%	0.2%
UC	1.72%	0.41%	0.0%
Firefox	0.49%	0.30%	0.7%
Others	1.42%	7.74%	15.3%

### Usage share of tablet browsers

Browser	Statcounter <sup>[25]</sup> September 2020	NetMarketShare <sup>[26]</sup> September 2020
Safari	46.86%	46.40%
Chrome	39.77%	44.82%
AOSP	11.42%	2.45%
Samsung Internet	N/A	3.53%
Opera	0.62%	0.63%
Firefox	0.30%	0.19%
Others	1.03%	1.98%

# Magnificent 3 – really or even 2



# Agenda

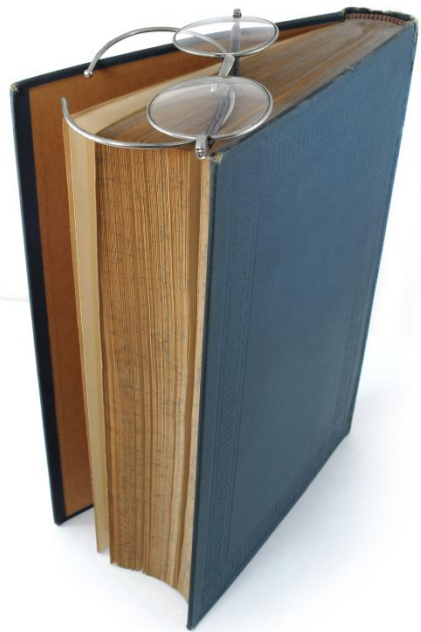
- About Course
- Automation reminder
- Automation types
- Basic concepts of web applications
- **What is Website?**
- DOM
- HTML, CSS, JS - Basics
- Why Java?
- Java variables
- Practice

# So what is website?





# What is a webpage?



A **web page** or **webpage** is a document commonly written in HyperText Markup Language (HTML) that is accessible through the Internet or other network using an Internet browser. A web page is accessed by entering a URL address and may contain text, graphics, and hyperlinks to other web pages and files.

SitePoint Pty Ltd [AU] | https://www.sitepoint.com/how-to-use-push-notifications-for-web-applications/

☆

7

8

R

K

Web Push

SendPulse

Learning Hubs

Books

Videos

Email Courses

Login

Sign Up

www.sitepoint.com wants to

Show notifications

Allow

Block

196 Shares

f 171

G+

in

# Web Push Notifications Protocol

The web Push Notifications protocol is relatively new. It gives web applications the ability to act as native applications and receive messages pushed to them from a server at any time even when the web app is not active or not currently loaded in a browser. This lets you engage users with urgent and relevant notifications when they are not using your application and motivate them to return to the application.

Web Push Message

## We value your privacy

We use cookies to provide you the best possible experience of SitePoint. Want to know more? Read our [Terms of Service](#) and [privacy policy](#). Have questions? Please contact [support@sitepoint.com](mailto:support@sitepoint.com), we're happy to help!

[Terms of Services](#)

|

[Privacy Policy](#)

I DO NOT ACCEPT

I ACCEPT

[Show Purposes](#)

Powered by [Quantcast](#)

# What is GUI?

There are two types of interfaces for a computer application. Command Line Interface is where you type text and computer responds to that command. GUI stands for Graphical User Interface where you interact with the computer using images rather than text.

Following are the GUI elements which can be used for interaction between the user and application:

☐ True    ☐ False    *Radio Button*

☐ Check    ☐ Check    *Check Box*

*Text Box*

*List Box*

GUI Testing is a validation of the above elements.

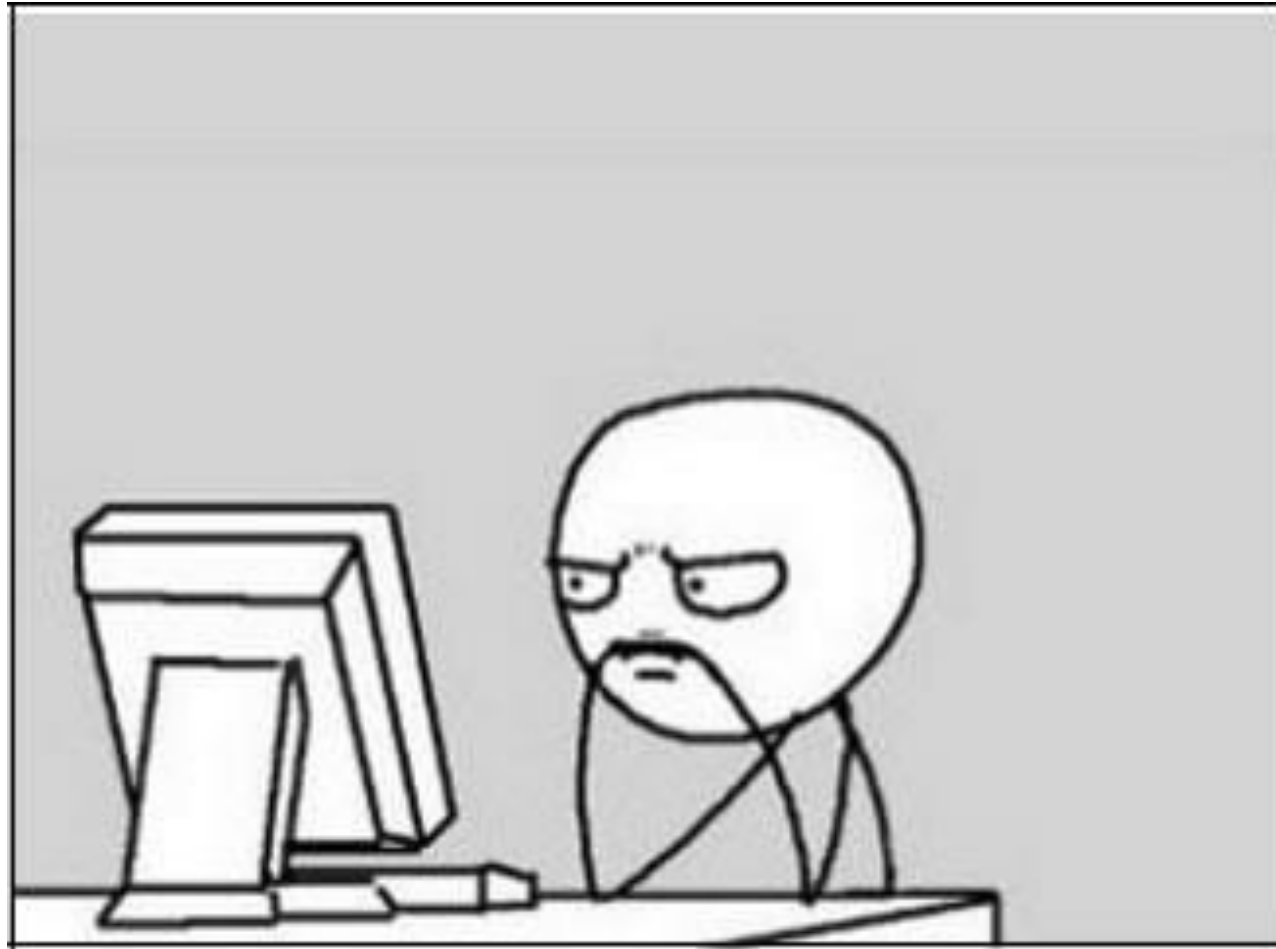
# What do you Check-in GUI Testing?

The following checklist will ensure detailed GUI Testing in Software Testing.

- Check all the GUI elements for size, position, width, length, and acceptance of characters or numbers. For instance, you must be able to provide inputs to the input fields.
- Check you can execute the intended functionality of the application using the GUI
- Check Error Messages are displayed correctly
- Check for Clear demarcation of different sections on screen
- Check Font used in an application is readable
- Check the alignment of the text is proper
- Check the Color of the font and warning messages is aesthetically pleasing
- Check that the images have good clarity
- Check that the images are properly aligned
- Check the positioning of GUI elements for different screen resolution.

# Not only GUI is important...

- To be able to test and analyze web, we need to analyze its content...
- How to do that?



# Tools to help to analyze web content for:

- Accessibility
- Marketing – keywords, google adwords
- HTML,CSS, JS – language validation
- Performance – speed of our web
- Basic user satisfaction

# Validation for Web Content

- <https://validator.w3.org/>

## Nu Html Checker

This tool is an ongoing experiment in better HTML checking, and its behavior remains subject to change

Showing results for <https://www.wobenzym.lv/>

Checker Input

Show ☐ source ☐ outline ☐ image report

Check by

<https://www.wobenzym.lv/>

Use the Message Filtering button below to hide/show particular messages, and to see total counts of errors and warnings.

1. **Warning** The `type` attribute is unnecessary for JavaScript resources.  
From line 1, column 609; to line 1, column 639  
`zym.lv/'/><script type="text/javascript"> (func`
2. **Warning** The `type` attribute for the `style` element is not needed and should be omitted.  
From line 1, column 1898; to line 1, column 1920  
`con.ico'/'><style type='text/css'>articl`

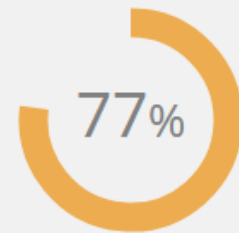
# Web analyzes

<https://www.dareboost.com/>

<http://www.wobenzym.lv>

Report generated on Apr 18, 2019 8:07:01 PM

[Download report](#)



Basics  
are almost there...



Issues



Improvements



Successes



[See your priorities](#) ▾

SIMULATED VISITOR: Chrome Washington DC 10.0/2.0Mbps (Latency: 28 ms)

[Edit](#)

Requests



HTML CSS Scripts Images Others

Weight



First Byte



Start Render




Fully loaded





# Web overall analyses

<http://nibbler.silktide.com/>

 nibbler

HomeAboutPro version

## Report for www.wobenzym.lv

8.0

**Overall**  
The overall score for this website.

9.1


**Accessibility**  
How accessible the website is to mobile and disabled users.  
[See contributing tests](#)

7.2

**Experience**  
How satisfying the website is likely to be for users.  
[See contributing tests](#)

5.3

**Marketing**  
How well marketed and popular the website is.  
[See contributing tests](#)



SIA "Mucos-Balt" piedāvā  
Zāļu nepamatota lietošana ir kaitīga veselībai. Zāļu nepareiza lietošana ir kaitīga veselībai.

Nibbler tested a sample of 5 pages from this website at 7:56 PM on Apr 18, 2019 (EEST).

### Overview

Twitter	0.0
Popularity	0.0
Printability	0.0
Incoming links	4.1
Internal links	4.4
Social interest	5.9
Facebook page	7.5
Amount of content	8.6
URL format	10
Meta tags	10
Images	10
Headings	10
Page titles	10
Analytics	10
Server behavior	10


# Website cookies

**Cookies**

This site uses cookies to offer you a better browsing experience. Find out more on [how we use cookies and how you can change your settings](#).

[I accept cookies](#) [I refuse cookies](#)

A-Z Index | Sitemap | About this site | FAQ | What's New | Legal notice | **Cookies** | Contact | Search English (en)

 **European Commission**

European Commission > Site with more first party cookies

Example of a functionality/item storing cookies:

**3** This part of the page requires cookies. [I accept](#) / [I refuse](#) this site's cookies.

Last update: DD/MM/YYYY | [Top](#)

# What is Cookie

---

- **A cookie is some small information stored in a text file on the user's hard drive by a web server**
- **Later used by the web browser to retrieve information from that machine**
- **Instructions for reading and writing cookies are coded by website authors and executed by user browsers**

# Web user tracking

## cookies



- **session vs. persistent cookies**
  - a session cookie is stored only while the user is connected to the particular Web server – the cookie is deleted when the user disconnects
  - persistent cookies are set to expire at some point in the future
    - many are set to expire a number of years forward

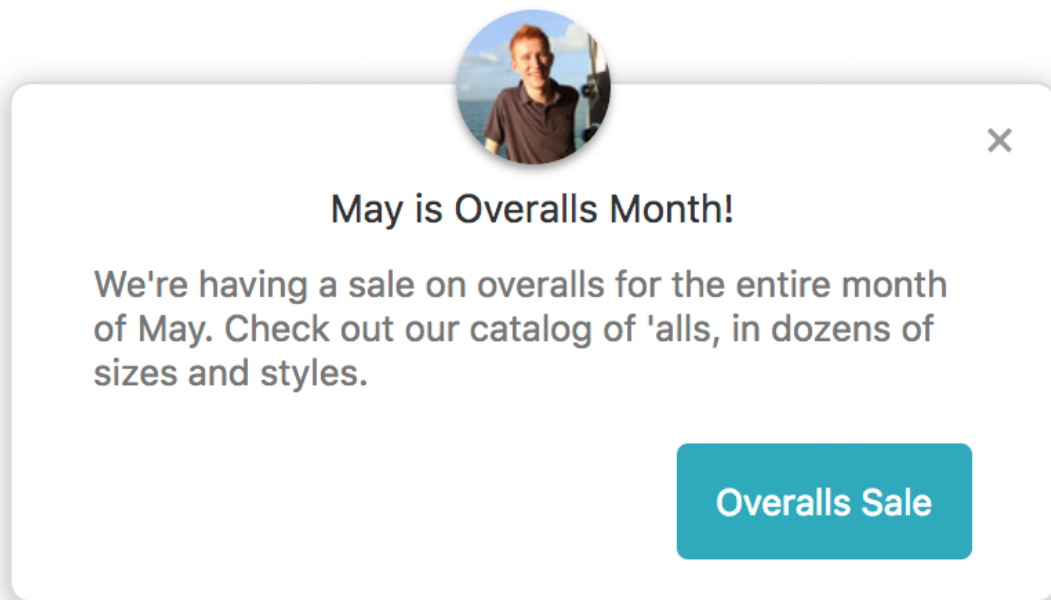
# Web cache

## **Example 1**

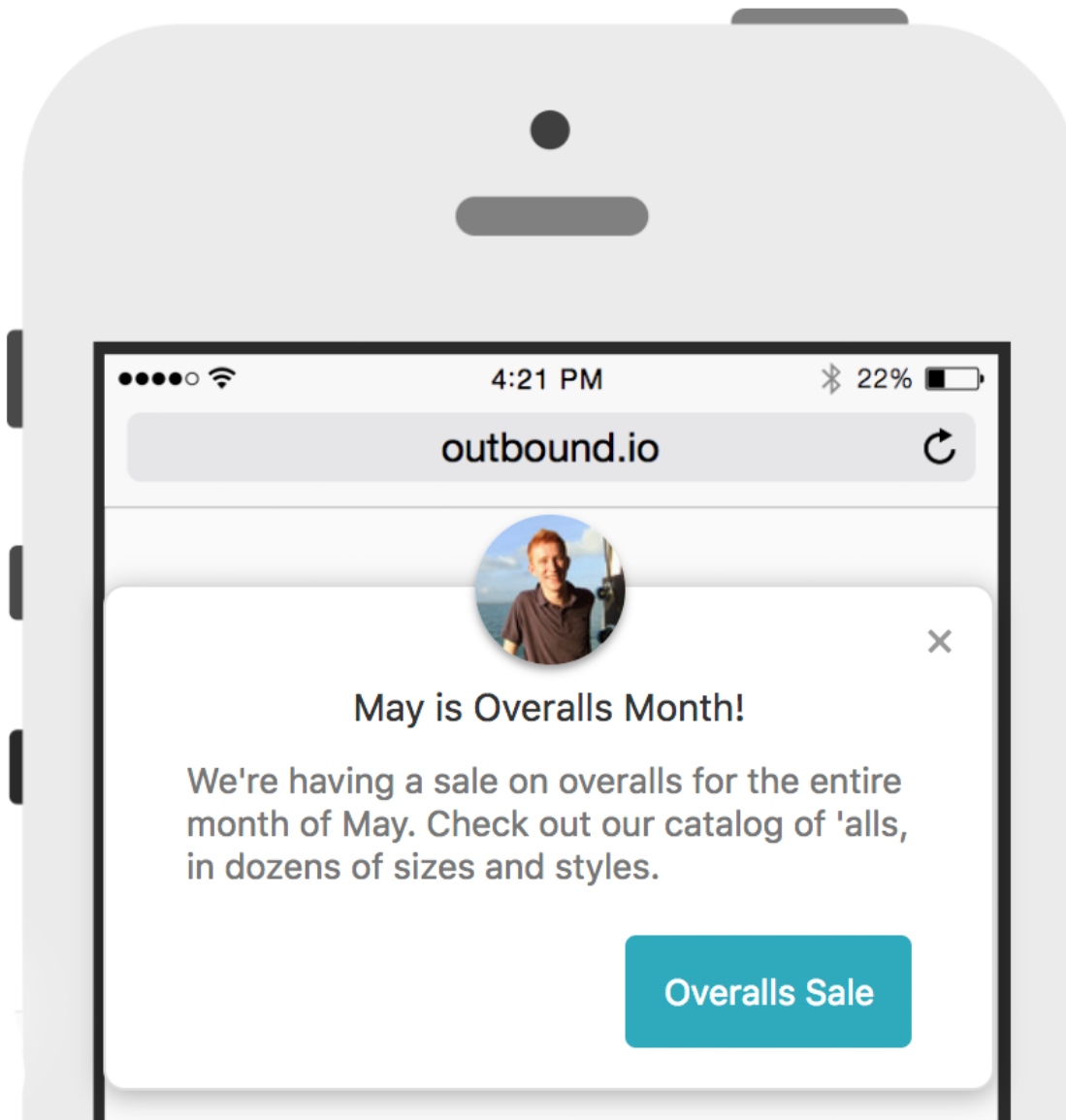
- Load site 1a.lv, save loading time
- See how many items are from cache
- Delete cache
- Load site 1a.lv again
- Compare loading times - slower

# Web notifications





# Desktop



# Mobile

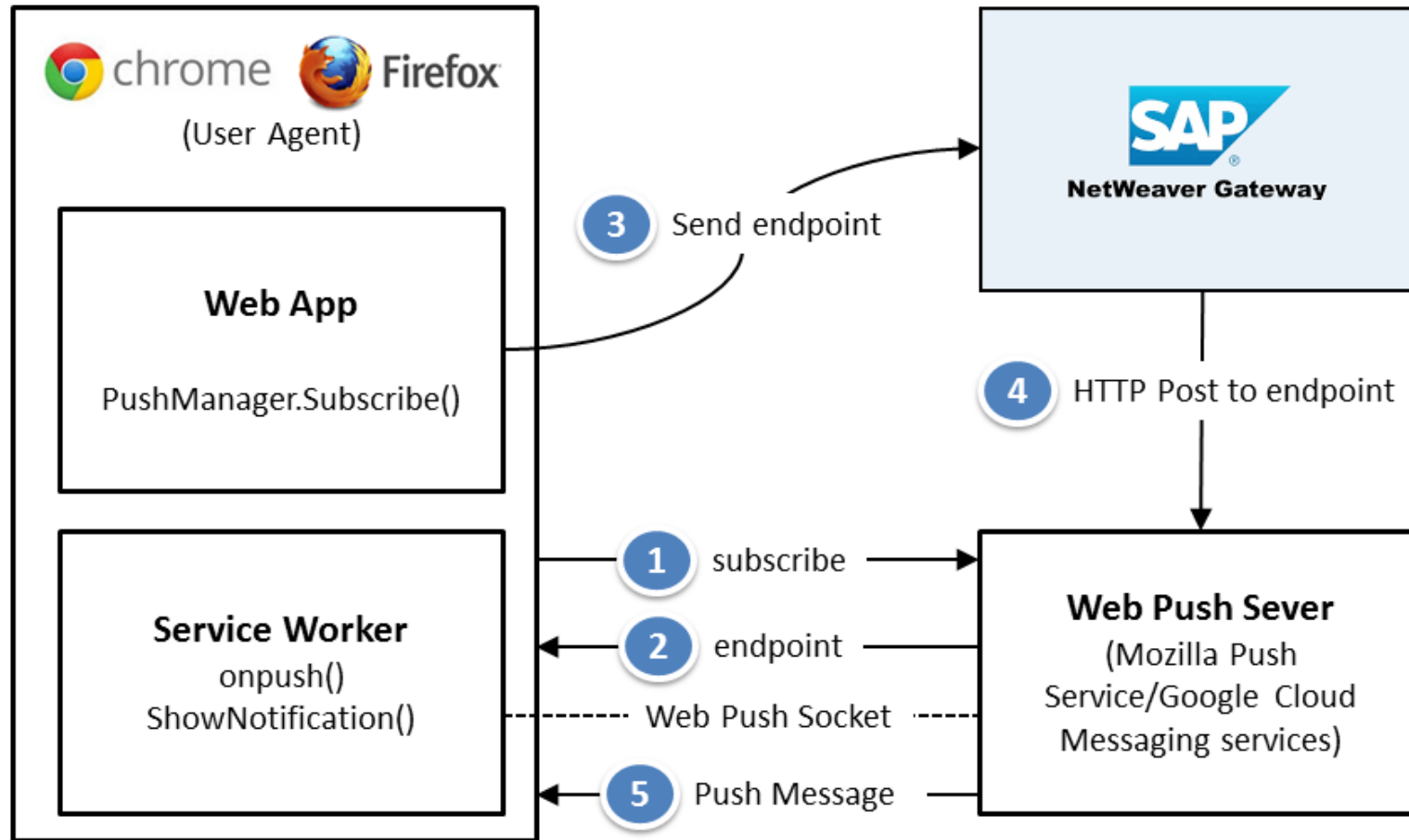


# Notifications





# Notifications



## ← Notifications

Ask before sending (recommended)



Block

Add



<https://coub.com>:443



<https://www.facebook.com>:443



<https://www.javacodegeeks.com>:443



<https://en.softonic.com>:443



<https://www.ultimateqa.com>:443



<https://www.youtube.com>:443



<https://www.novatours.lv>:443



# Website

Indian Railway Catering and Tourism Corporation Limited  
A Government of India Enterprise

Centre For Railway Information Systems  
(An organisation of the Ministry of Railways, Govt. of India)

Home | Contact Us |

Launch Win an iPhone 7 every month on IRCTC 588 Platinum Card !! Apply Now !! For offer details - Click here !! Lucky Draw Winners for Advertise with us

Book Meal | Tourism | Flights | Co-Branded Cards | Alerts & Updates | More | NEW Lucky Draw | NEW Wi-Fi Railway Stations | NEW Counter Ticket Cancellation

Login

User ID: 583853 | Set Up | Password: | Forgot Password | Captcha: J9Z4UV | Captcha letters are case sensitive and to be entered in Upper Case only | Request OTP | Login | IRCTC Agent Link

**All Out IT'S GOOD TO BE TOUGH.**

I-Ticket services through IRCTC website have been discontinued w.e.f. 01.03.2018. || Booking of Mumbai Suburban Se:

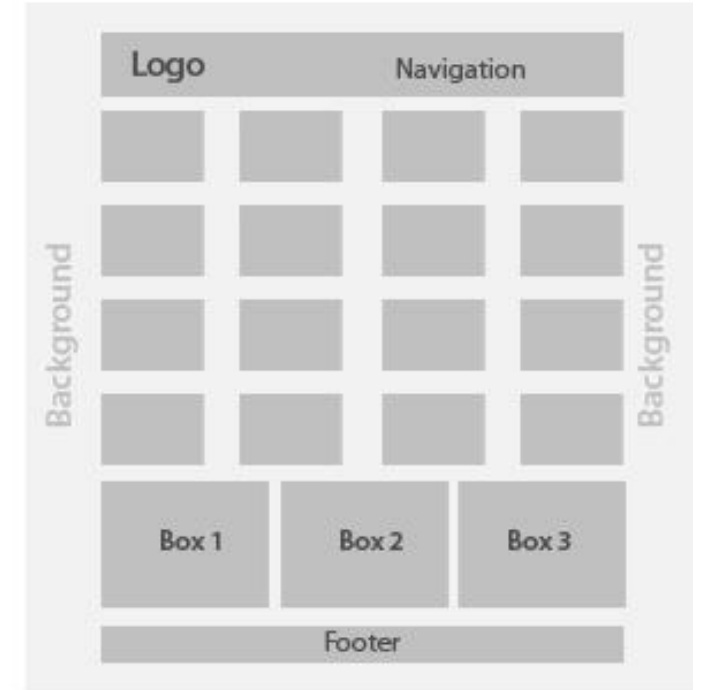
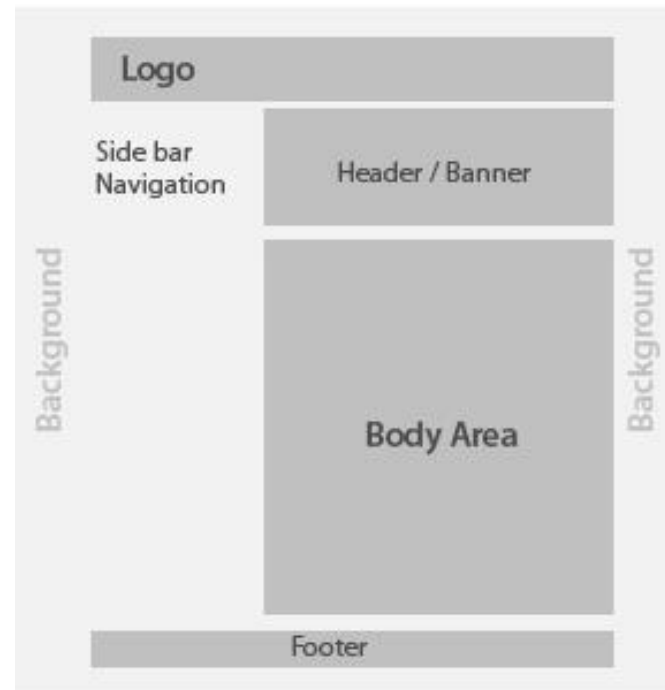
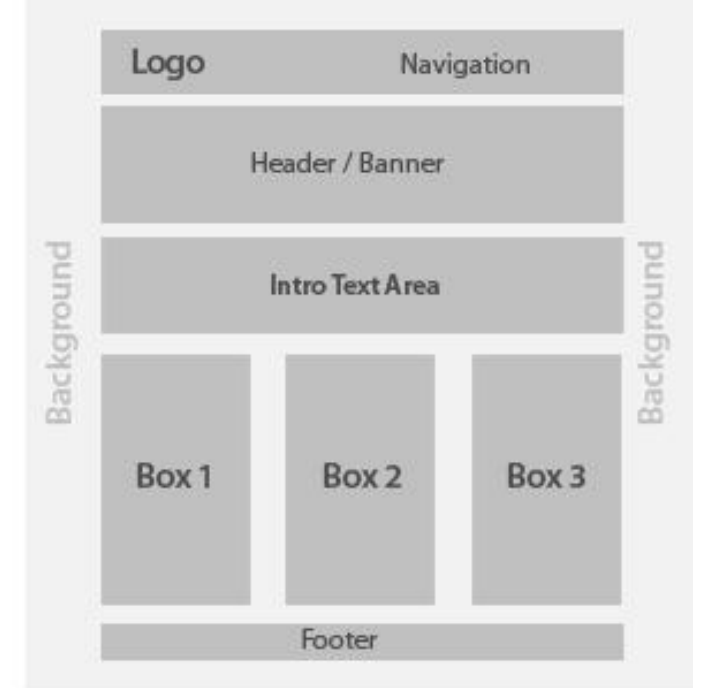
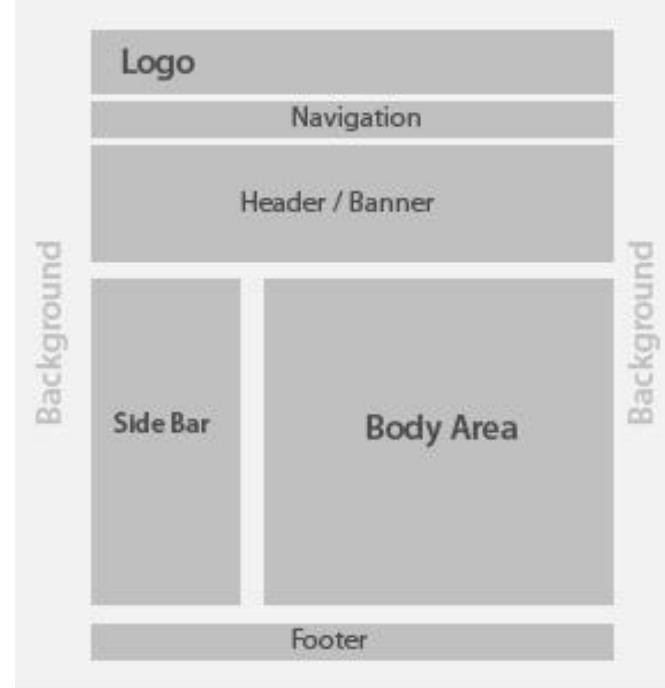
IRCTC Official App	IRCTC Zone	I-Ticket
User Guide	IRCTC Prepaid	IRCTC Loyalty Program
Enquiries	Mobile Zone	Advertise with us
General Information	Train/Coach Booking (FTR)	Integration Policy
Refund Rule	Agents	Claim Your Offer

**BUY ONE GET ONE FREE** Book a movie ticket at BookMyShow and get the second ticket free with your HSBC Credit Card. **Apply now**

HSBC

Copyright © 2018 - www.irctc.co.in. All Rights Reserved  
Designed and Hosted by CRIS  
[Compatible Browser](#)

# Web Page Layout



# WEB page



## WEB Page

### HTML

#### Content and Structure

Headings  
Paragraphs  
Lists

### CSS

#### Presentation

Font  
Color  
Margins

### JavaScript

#### Behavior

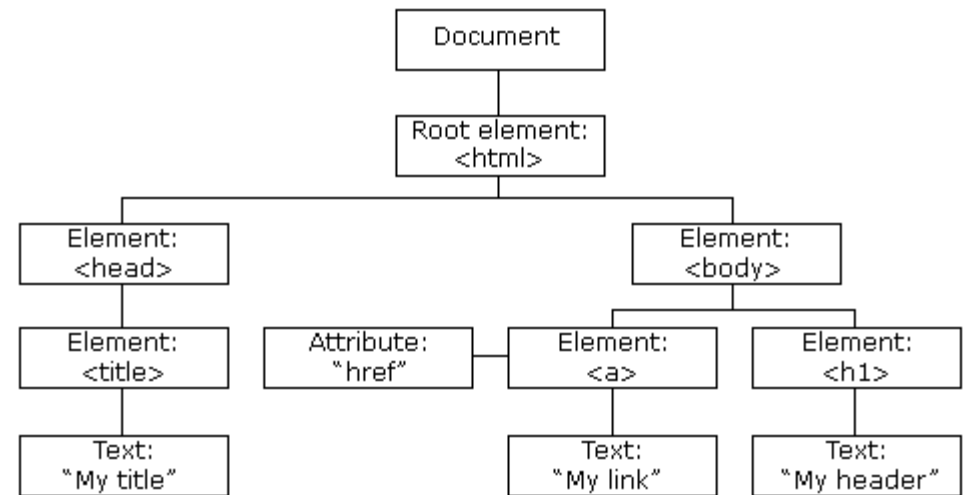
Dynamic display  
User interaction  
Popups

# Agenda

- About Course
- Automation reminder
- Automation types
- Basic concepts of web applications
- What is Website?
- **DOM**
- HTML, CSS, JS - Basics
- Why Java?
- Java variables
- Practice

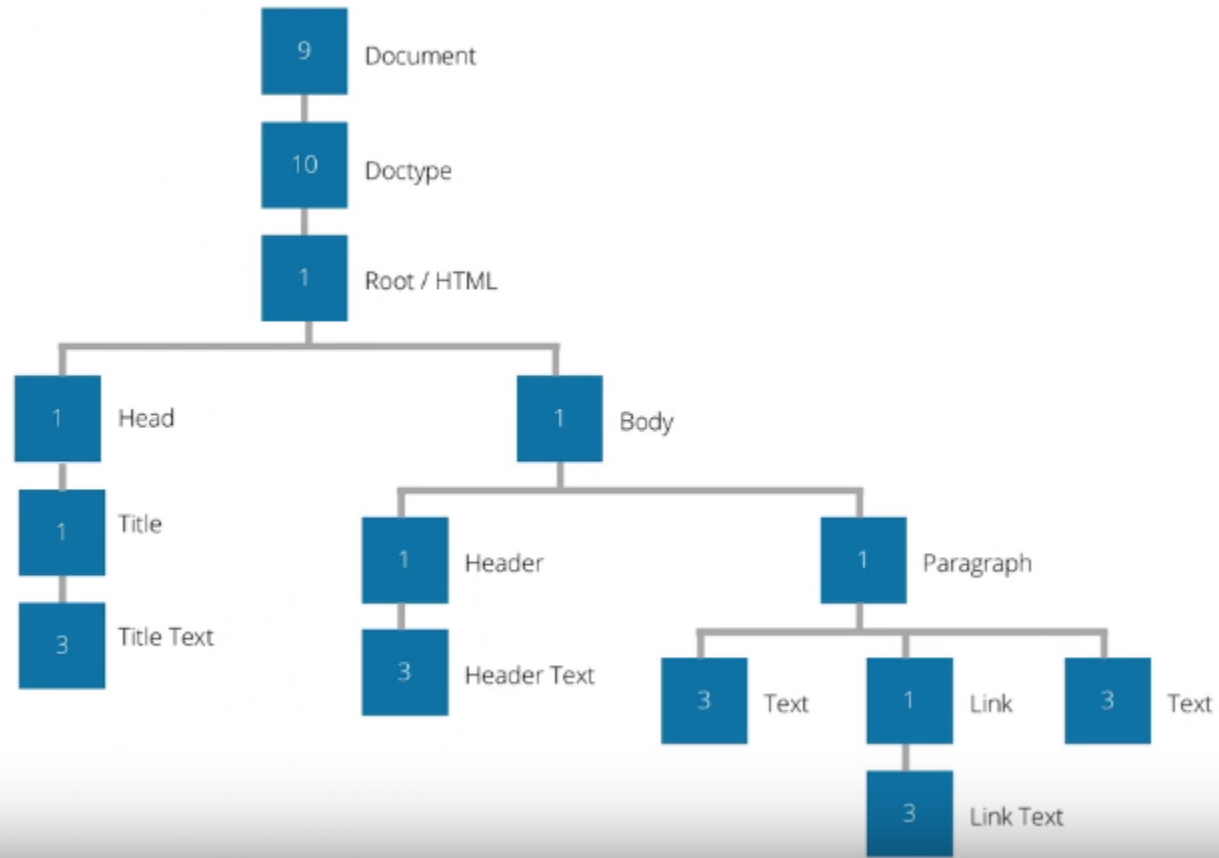
# The HTML DOM (Document Object Model)

- When a web page is loaded, the browser creates a Document Object Model of the page
- The HTML DOM model is constructed as a tree of Objects
- In other words: The HTML DOM is a standard for how to get, change, add, or delete HTML elements
- With the document object model, JavaScript gets all the power it needs to create dynamic HTML
- Document Object Model or DOM is an essential component of web development and **automation**



# HTML DOM example

```
<!DOCTYPE html>
<html>
<head>
  <title>The DOM</title>
</head>
<body>
  <h1>Title</h1>
  <p>Lorem <a title="Learn more"
    href="#">to the</a> ipsum</p>
</body>
</html>
```





# Agenda

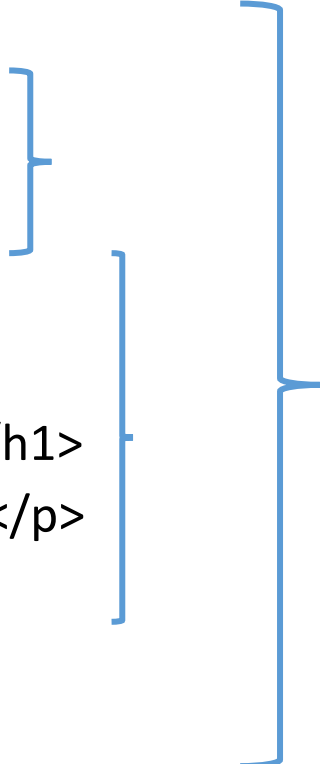
- About Course
- Automation reminder
- Automation types
- Basic concepts of web applications
- What is Website?
- DOM
- **HTML, CSS, JS - Basics**
- Why Java?
- Java variables
- Practice

# HTML (Hyper Text Markup Language)

```
<!DOCTYPE html>
<html>
<head>
<title>Page Title</title>
</head>
<body>

<h1>This is a Heading</h1>
<p>This is a paragraph.</p>

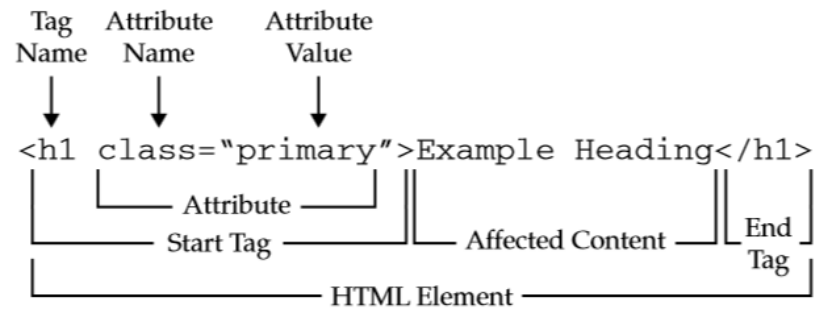
</body>
</html>
```



- HTML describes the structure of Web pages using markup
- HTML elements are the building blocks of HTML pages
- HTML elements are represented by tags
- HTML tags label pieces of content such as "heading", "paragraph", "table", and so on
- Browsers do not display the HTML tags, but use them to render the content of the page
- All HTML documents must start with a document type declaration: `<!DOCTYPE html>`
- Begins with `<html>` and ends with `</html>`
- The visible part of the HTML document is between `<body>` and `</body>`

# HTML syntax overview

A graphical overview of the HTML markup syntax shown so far is presented here:



Tag	Description
<code>&lt;html&gt; ... &lt;/html&gt;</code>	Declares the Web page to be written in HTML
<code>&lt;head&gt; ... &lt;/head&gt;</code>	Delimits the page's head
<code>&lt;title&gt; ... &lt;/title&gt;</code>	Defines the title (not displayed on the page)
<code>&lt;body&gt; ... &lt;/body&gt;</code>	Delimits the page's body
<code>&lt;h n&gt; ... &lt;/h n&gt;</code>	Delimits a level <i>n</i> heading
<code>&lt;b&gt; ... &lt;/b&gt;</code>	Set ... in boldface
<code>&lt;i&gt; ... &lt;/i&gt;</code>	Set ... in italics
<code>&lt;center&gt; ... &lt;/center&gt;</code>	Center ... on the page horizontally
<code>&lt;ul&gt; ... &lt;/ul&gt;</code>	Brackets an unordered (bulleted) list
<code>&lt;ol&gt; ... &lt;/ol&gt;</code>	Brackets a numbered list
<code>&lt;li&gt; ... &lt;/li&gt;</code>	Brackets an item in an ordered or numbered list
<code>&lt;br&gt;</code>	Forces a line break here
<code>&lt;p&gt;</code>	Starts a paragraph
<code>&lt;hr&gt;</code>	Inserts a horizontal rule
<code>&lt;img src="..."&gt;</code>	Displays an image here
<code>&lt;a href="..."&gt; ... &lt;/a&gt;</code>	Defines a hyperlink

# CSS (Cascading Style Sheets)

- `body {`
- `background-color: lightblue;`
- `}`

- `h1 {`
- `color: white;`
- `text-align: center;`
- `}`

- `p {`
- `font-family: verdana;`
- `font-size: 20px;`
- `}`

- **CSS** stands for **Cascading Style Sheets**
- CSS describes **how HTML elements are to be displayed**
- Contains the rules for the presentation of HTML.
- CSS was introduced to keep the presentation information separate from HTML markup (content).
- CSS **saves a lot of work**. It can control the layout of multiple web pages all at once
- External stylesheets are stored in **CSS files**



HTML

+



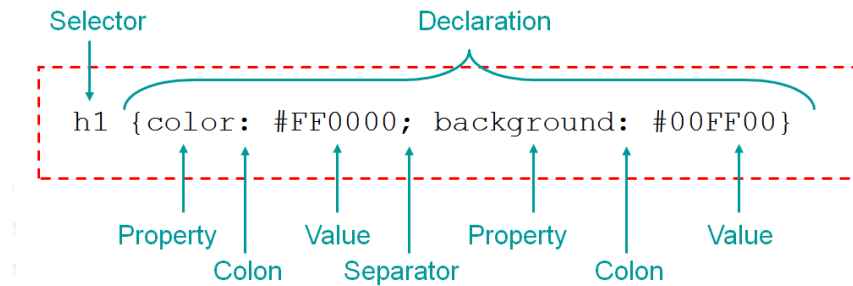
CSS

=



Web Page

# CSS syntax overview



The element Selector

```
p {  
  text-align: center;  
  color: red;  
}
```

The class Selector

```
.className {  
  text-align: center;  
  color: red;  
}
```

The id Selector

```
#myId {  
  text-align: center;  
  color: red;  
}
```

Grouping Selectors

```
h1, h2 {  
  text-align: center;  
  color: red;  
}
```

Property	Description	Example
color	Sets the foreground color of an element	body{color: #FCC9814;}
background-color	Sets the background color of an element	body{background-color: green;}
background-image	Inserts a background image	body{background-image: url ("earth.gif");}
background-repeat	Specifies how background image will repeat itself. It may repeat horizontally (repeat-x) vertically (repeat-y) or (repeat) or may not be repeated (no-repeat)	body{background-repeat: repeat-x;}

# JavaScript

```
<!DOCTYPE html>
<html>
<body>

<h2>My First JavaScript</h2>

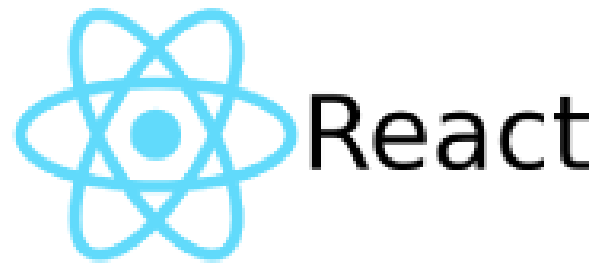
<button type="button"
onclick="document.getElementById('demo').innerHTML =
Date()">
Click me to display Date and Time.</button>

<p id="demo"></p>

</body>
</html>
```

- JavaScript is the programming language of HTML and the Web
- JavaScript Can Change HTML Content
- JavaScript is an interpreted language. Thus, it doesn't need to be compiled

# Popular JavaScript Frameworks



# JavaScript structure

```
<p id="demo"></p>
```

```
<script>
```

```
document.getElementById("demo").innerHTML = "My First JavaScript";
```

```
</script>
```

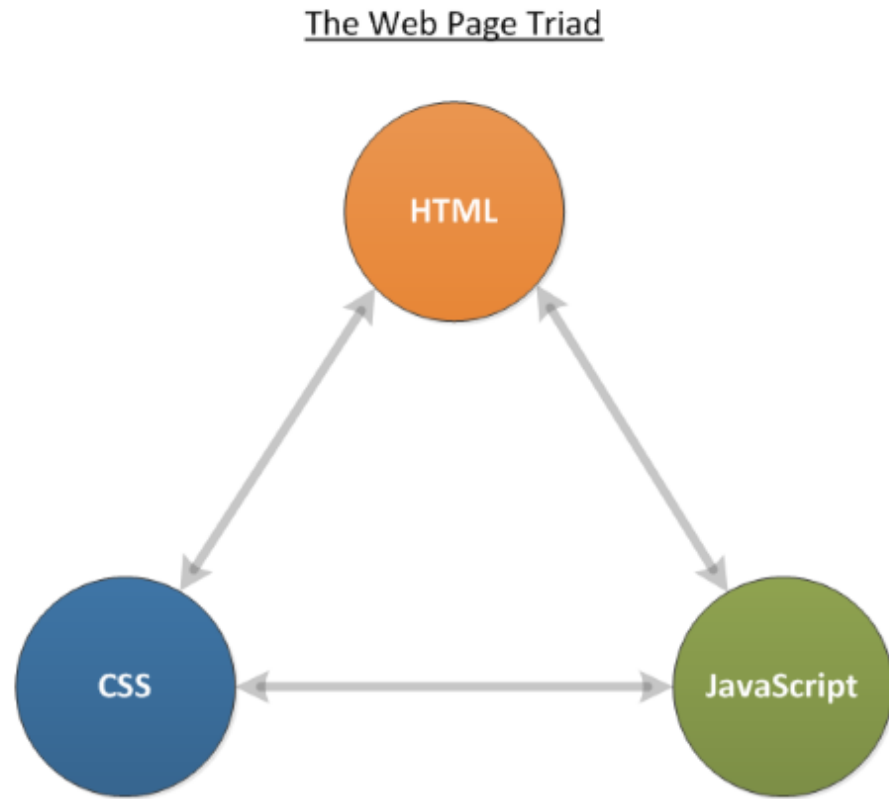
```
<button onclick="document.getElementById('myImage').src='pic_bulbon.gif'">Turn  
on the light</button>
```

```

```



# Summary



- HTML is the markup language that we use to structure and give meaning to our web content, for example defining paragraphs, headings, and data tables, or embedding images and videos in the page.
- CSS is a language of style rules that we use to apply styling to our HTML content, for example setting background colors and fonts, and laying out our content in multiple columns.
- JavaScript is a scripting language that enables you to create dynamically updating content, control multimedia, animate images, and pretty much everything else.

# Agenda

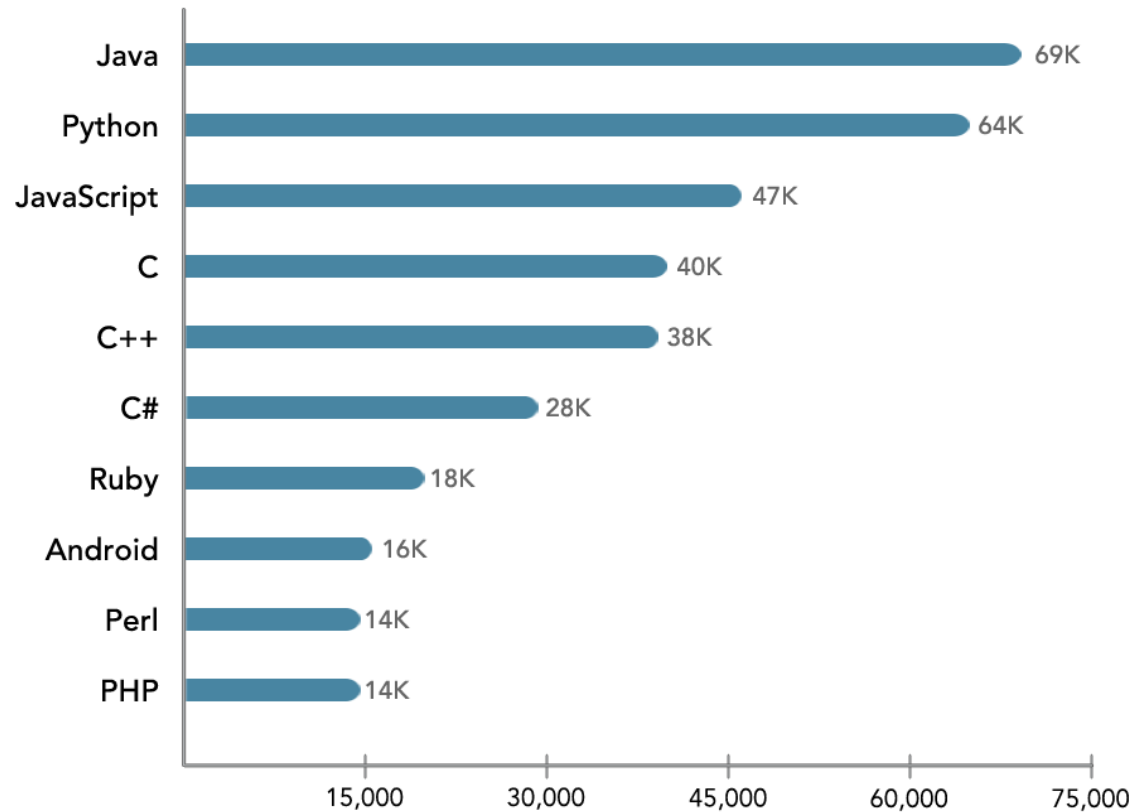
- About Course
- Automation reminder
- Automation types
- Basic concepts of web applications
- What is Website?
- DOM
- HTML, CSS, JS - Basics
- **Why Java?**
- Java variables
- Practice

# Why Choose Java?



# Most in-demand programming languages of 2019

*Based on Indeed.com job postings in the USA - Feb 1, 2019*



*Image Source: CodingNomads*

## Why Java?

# Why Java?

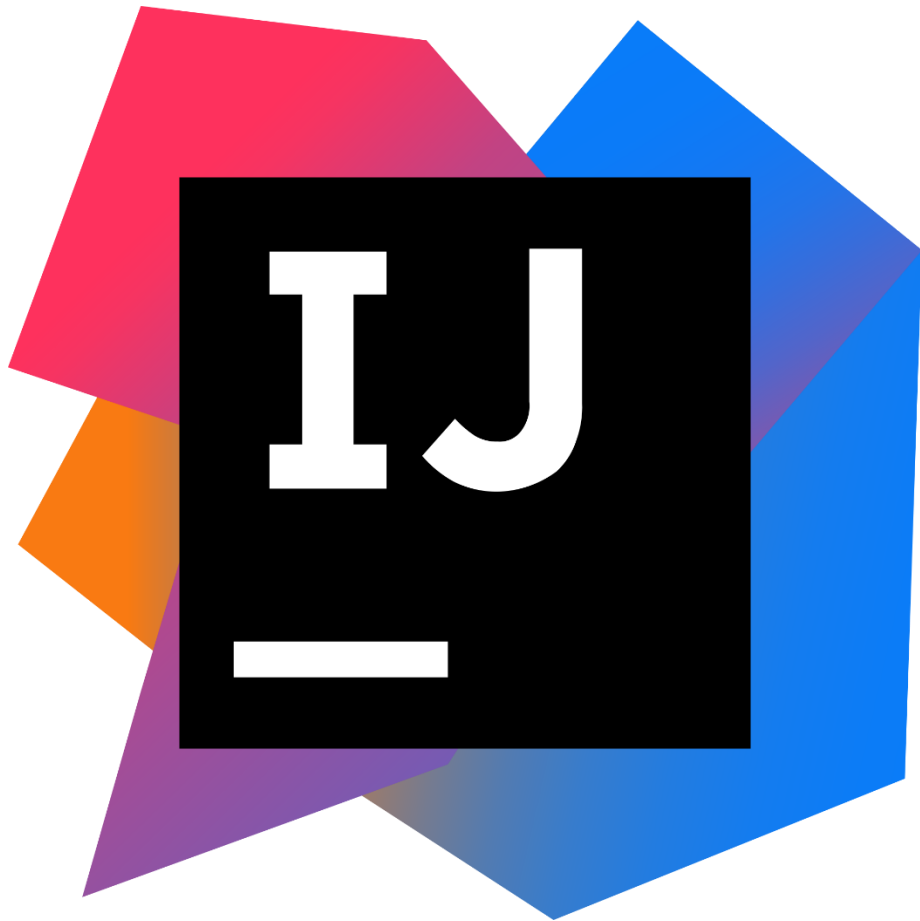
1. A lot of libraries for automation
2. Most common language for automation
3. Perfect on building and maintaining big projects
4. Multi module system support
5. To understand Java need basic OOP understanding
6. .... But Hard to master



# What will we use

- Java – work with IntelliJ
- Selenium Webdriver
- WebDriverManager
- Selenide
- Cucumber
- Gradle
- Git

IDEA







## DEFAULT KEYMAP



### Remember these Shortcuts

Smart code completion	<b>Ctrl + Shift + Space</b>
Search everywhere	<b>Double Shift</b>
Show intention actions and quick-fixes	<b>Alt + Enter</b>
Generate code	<b>Alt + Ins</b>
Parameter info	<b>Ctrl + P</b>
Extend selection	<b>Ctrl + W</b>
Shrink selection	<b>Ctrl + Shift + W</b>
Recent files popup	<b>Ctrl + E</b>
Rename	<b>Shift + F6</b>

### General

Open corresponding tool window	<b>Alt + #[0-9]</b>
Save all	<b>Ctrl + S</b>
Synchronize	<b>Ctrl + Alt + Y</b>
Toggle maximizing editor	<b>Ctrl + Shift + F12</b>
Inspect current file with current profile	<b>Alt + Shift + I</b>
Quick switch current scheme	<b>Ctrl + BackQuote (`)</b>
Open Settings dialog	<b>Ctrl + Alt + S</b>
Open Project Structure dialog	<b>Ctrl + Alt + Shift + S</b>
Find Action	<b>Ctrl + Shift + A</b>

### Debugging

Step over / into	<b>F8 / F7</b>
Smart step into / Step out	<b>Shift + F7 / Shift + F8</b>
Run to cursor	<b>Alt + F9</b>
Evaluate expression	<b>Alt + F8</b>
Resume program	<b>F9</b>
Toggle breakpoint	<b>Ctrl + F8</b>
View breakpoints	<b>Ctrl + Shift + F8</b>

### Search / Replace

Search everywhere	<b>Double Shift</b>
Find	<b>Ctrl + F</b>
Find next / previous	<b>F3 / Shift + F3</b>
Replace	<b>Ctrl + R</b>
Find in path	<b>Ctrl + Shift + F</b>
Replace in path	<b>Ctrl + Shift + R</b>
Select next occurrence	<b>Alt + J</b>
Select all occurrences	<b>Ctrl + Alt + Shift + J</b>
Unselect occurrence	<b>Alt + Shift + J</b>

—Productivity Boosters

### Editing

Basic code completion	<b>Ctrl + Space</b>
Smart code completion	<b>Ctrl + Shift + Space</b>
Complete statement	<b>Ctrl + Shift + Enter</b>
Parameter info (within method call arguments)	<b>Ctrl + P</b>
Quick documentation lookup	<b>Ctrl + Q</b>
External Doc	<b>Shift + F1</b>
Brief Info	<b>Ctrl + mouse</b>
Show descriptions of error at caret	<b>Ctrl + F1</b>
Generate code...	<b>Alt + Insert</b>
Override methods	<b>Ctrl + O</b>
Implement methods	<b>Ctrl + I</b>
Surround with...	<b>Ctrl + Alt + T</b>
Comment / uncomment with line comment	<b>Ctrl + /</b>
Comment / uncomment with block comment	<b>Ctrl + Shift + /</b>
Extend selection	<b>Ctrl + W</b>
Shrink selection	<b>Ctrl + Shift + W</b>
Context info	<b>Alt + Q</b>
Show intention actions and quick-fixes	<b>Alt + Enter</b>
Reformat code	<b>Ctrl + Alt + L</b>
Optimize imports	<b>Ctrl + Alt + O</b>
Auto-indent line(s)	<b>Ctrl + Alt + I</b>
Indent / unindent selected lines	<b>Tab / Shift + Tab</b>
Cut current line to clipboard	<b>Ctrl + X, Shift + Delete</b>
Copy current line to clipboard	<b>Ctrl + C, Ctrl + Insert</b>
Paste from clipboard	<b>Ctrl + V, Shift + Insert</b>
Paste from recent buffers...	<b>Ctrl + Shift + V</b>
Duplicate current line	<b>Ctrl + D</b>
Delete line at caret	<b>Ctrl + Y</b>
Smart line join	<b>Ctrl + Shift + J</b>
Smart line split	<b>Ctrl + Enter</b>
Start new line	<b>Shift + Enter</b>
Toggle case for word at caret or selected block	<b>Ctrl + Shift + U</b>
Select till code block end / start	<b>Ctrl + Shift + ] / [</b>
Delete to word end	<b>Ctrl + Delete</b>
Delete to word start	<b>Ctrl + Backspace</b>
Expand / collapse code block	<b>Ctrl + NumPad+ / -</b>
Expand all	<b>Ctrl + Shift + NumPad+</b>
Collapse all	<b>Ctrl + Shift + NumPad-</b>
Close active editor tab	<b>Ctrl + F4</b>

### Refactoring

Copy	<b>F5</b>
Move	<b>F6</b>
Safe Delete	<b>Alt + Delete</b>
Rename	<b>Shift + F6</b>
Refactor this	<b>Ctrl + Alt + Shift + T</b>
Change Signature	<b>Ctrl + F6</b>
Inline	<b>Ctrl + Alt + N</b>
Extract Method	<b>Ctrl + Alt + M</b>
Extract Variable	<b>Ctrl + Alt + V</b>
Extract Field	<b>Ctrl + Alt + F</b>
Extract Constant	<b>Ctrl + Alt + C</b>
Extract Parameter	<b>Ctrl + Alt + P</b>

### Navigation

Go to class	<b>Ctrl + N</b>
Go to file	<b>Ctrl + Shift + N</b>
Go to symbol	<b>Ctrl + Alt + Shift + N</b>
Go to next / previous editor tab	<b>Alt + Right / Left</b>
Go back to previous tool window	<b>F12</b>

Go to editor (from tool window)	<b>Esc</b>
Hide active or last active window	<b>Shift + Esc</b>
Go to line	<b>Ctrl + G</b>
Recent files popup	<b>Ctrl + E</b>
Navigate back / forward	<b>Ctrl + Alt + Left / Right</b>
Navigate to last edit location	<b>Ctrl + Shift + Backspace</b>
Select current file or symbol in any view	<b>Alt + F1</b>
Go to declaration	<b>Ctrl + B, Ctrl + Click</b>
Go to implementation(s)	<b>Ctrl + Alt + B</b>
Open quick definition lookup	<b>Ctrl + Shift + I</b>
Go to type declaration	<b>Ctrl + Shift + B</b>
Go to super-method / super-class	<b>Ctrl + U</b>
Go to previous / next method	<b>Alt + Up / Down</b>
Move to code block end / start	<b>Ctrl + ] / [</b>
File structure popup	<b>Ctrl + F12</b>
Type hierarchy	<b>Ctrl + H</b>
Method hierarchy	<b>Ctrl + Shift + H</b>
Call hierarchy	<b>Ctrl + Alt + H</b>
Next / previous highlighted error	<b>F2 / Shift + F2</b>
Edit source / View source	<b>F4 / Ctrl + Enter</b>
Show navigation bar	<b>Alt + Home</b>
Toggle bookmark	<b>F11</b>
Toggle bookmark with mnemonic	<b>Ctrl + F11</b>
Go to numbered bookmark	<b>Ctrl + #[0-9]</b>
Show bookmarks	<b>Shift + F11</b>

### Compile and Run

Make project	<b>Ctrl + F9</b>
Compile selected file, package or module	<b>Ctrl + Shift + F9</b>
Select configuration and run / debug	<b>Alt + Shift + F10 / F9</b>
Run / Debug	<b>Shift + F10 / F9</b>
Run context configuration from editor	<b>Ctrl + Shift + F10</b>

### Usage Search

Find usages / Find usages in file	<b>Alt + F7 / Ctrl + F7</b>
Highlight usages in file	<b>Ctrl + Shift + F7</b>
Show usages	<b>Ctrl + Alt + F7</b>

### VCS / Local History

Commit project to VCS	<b>Ctrl + K</b>
Update project from VCS	<b>Ctrl + T</b>
Push commits	<b>Ctrl + Shift + K</b>
'VCS' quick popup	<b>Alt + BackQuote (`)</b>

### Live Templates

Surround with Live Template	<b>Ctrl + Alt + J</b>
Insert Live Template	<b>Ctrl + J</b>



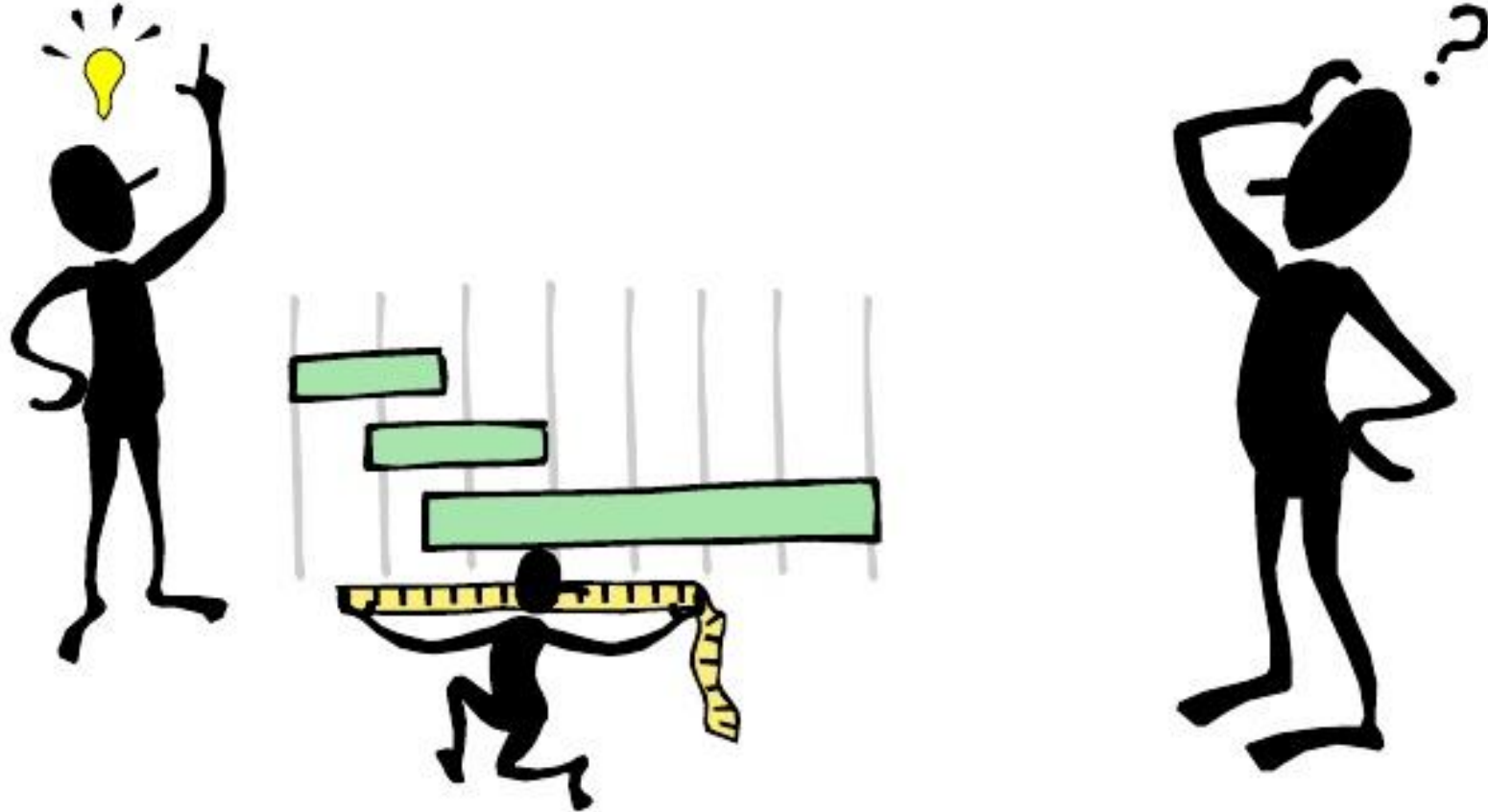
# IDEA Important key shortcut

- Ctrl+Alt+L – Format your code
- Ctrl+/ - comment line
- Shift+F6 – Refactor -> Rename something in the whole file
- During Debug – on breakpoint – choose line and press Alt+F8 – Evaluate expression

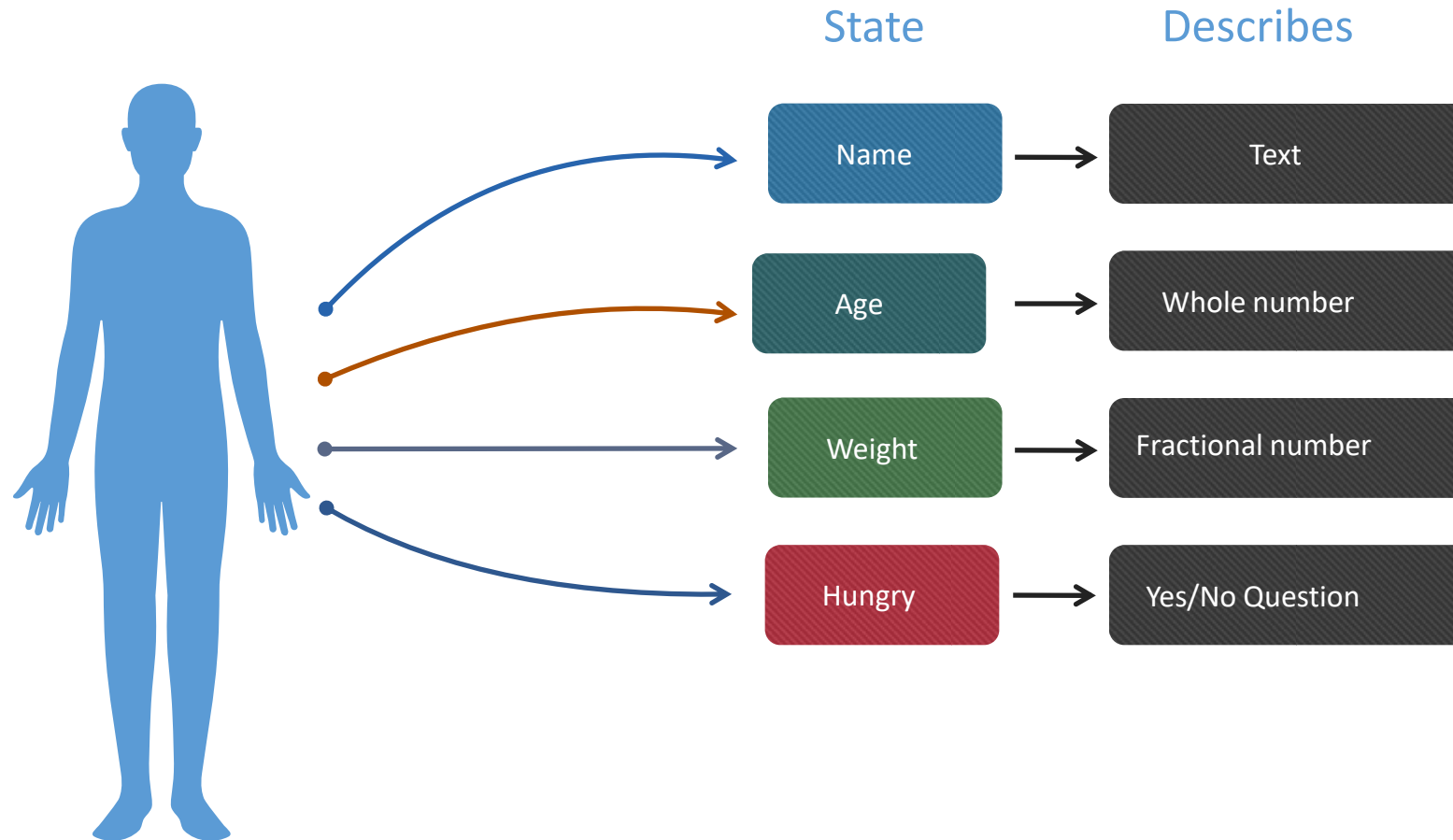
# Agenda

- About Course
- Automation reminder
- Automation types
- Basic concepts of web applications
- What is Website?
- DOM
- HTML, CSS, JS - Basics
- Why Java?
- **Java variables**
- Arithmetic Operators
- Practice

# What is a Variable?



# Variables relationship with objects

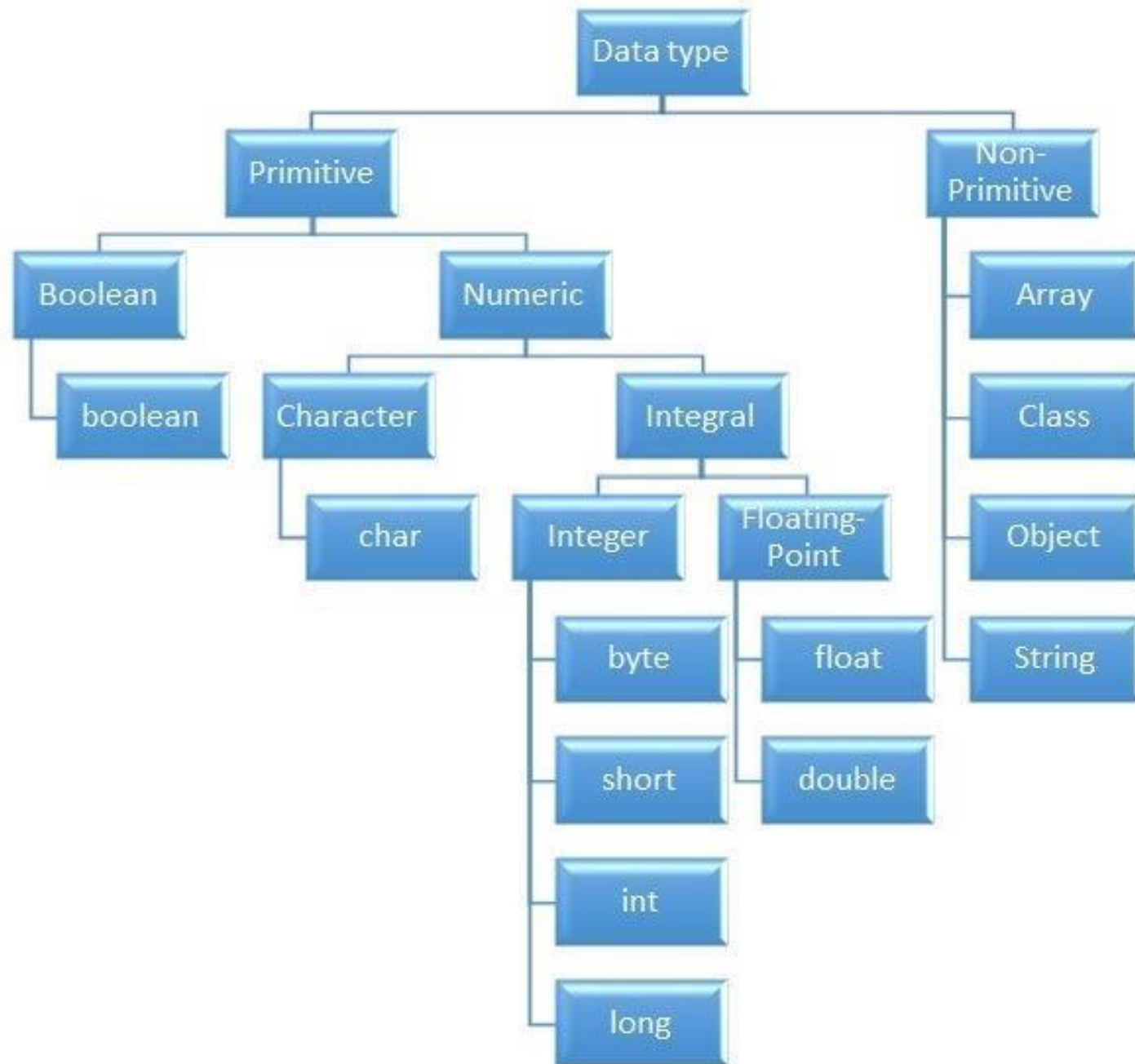


# Variables definition

- ▶ Variable is a **named placeholder** that
  - ▶ Stores **data**
  - ▶ Describes what **type of data** you can store
  - ▶ Describes **size** or **amount of data** it can store

# Data type categories

- ▶ Primitive values
  - ▶ Integer: *byte, short, int, long* (e.g. 3, 7, 42, 2018)
  - ▶ Fractional: *float, double* (e.g. 3.1415, 2.7, 19.0)
  - ▶ Logical: *boolean* (true or false)
  - ▶ Textual: *char* (e.g. a, b, c, x, y, z)
- ▶ Reference values
  - ▶ Everything else



# Primitive data types in depth: integer

Name	Assignable Values	Space
byte	-128 ... 127	1 byte
short	-32,768 ... 32,767	2 bytes
int	$-2^{31} \dots 2^{31} - 1$	4 bytes
long	$-2^{63} \dots 2^{63} - 1$	8 bytes



# Primitive data types in depth: floating point

Name	Precision	Space
float	Single	4 bytes
double	Double	8 bytes

# Primitive data types in depth: logical

Name	Assignable Values	Space
boolean	true / false	1 byte

# Primitive data types in depth: textual

Name	Assignable Values	Space
char (unicode)	0 ('\u0000') ... 65535 ('\uffff')	2 bytes

# VARIABLE DECLARATION IN JAVA: SYNTAX

- ▶ Variable declaration **without** value assignment

```
type name;
```

- ▶ Variable declaration **with** value assignment

```
type name = value;
```

# Variable declaration in Java: Example

- ▶ Variable declaration **without** value assignment

```
int age;
```

- ▶ Variable declaration **with** value assignment

```
int age = 22;
```

# Variable declaration breakdown

Variable data type

Assignment operator

**int** age = 22; ← End of statement

Variable value

Variable name

# Naming Rules

- ▶ Any variable is **allowed** to **start** with
  - ▶ Letters (**A-Z**)
  - ▶ Special characters ('\$' - dollar, '\_' - underscore)
- ▶ Any variable name is **allowed** to **contain**
  - ▶ Alphanumeric characters (**A-Z, 0-9**)
  - ▶ Special characters ('\$' - dollar, '\_' - underscore)
- ▶ Variable name is **case-sensitive**
- ▶ Java language **keywords**<sup>1</sup> or **reserved** words **cannot** be used as variable name

<sup>1</sup> List of keywords can be found at [https://docs.oracle.com/javase/tutorial/java/nutsandbolts/\\_keywords.html](https://docs.oracle.com/javase/tutorial/java/nutsandbolts/_keywords.html)

# Naming dos

- ▶ Single-worded name should be lowercase
- ▶ Multi-worded name should
  - ▶ First word lowercase
  - ▶ Subsequent words start with capital letters
  - ▶ No intervening spaces or punctuation
- ▶ Explains the purpose of variable



# Naming don'ts

- ▶ Starting variable name with \$ or \_ is highly discouraged
- ▶ Avoid using \$ anywhere in the variable name

# Naming dos and don'ts examples

- ▶ Please, **do**

- ▶ size, xCoordinate, skinColor, currentDayOfTheWeek

- ▶ Please, **don't**

- ▶ \_counter, \$bankBalance, Timestamp, 7daysOfTheWeek, !variableName, \*notPointer

# More examples

```
byte numberOfWheels = 4;  
short selfEsteem = -1;  
int studentsGraduated = 1001;  
long height = 80;  
float pie = 3.14f;  
double weight = 70.5;  
boolean hungry = true;  
char lastLetterOfTheAlphabet = 'Z';
```

# More Naming rules

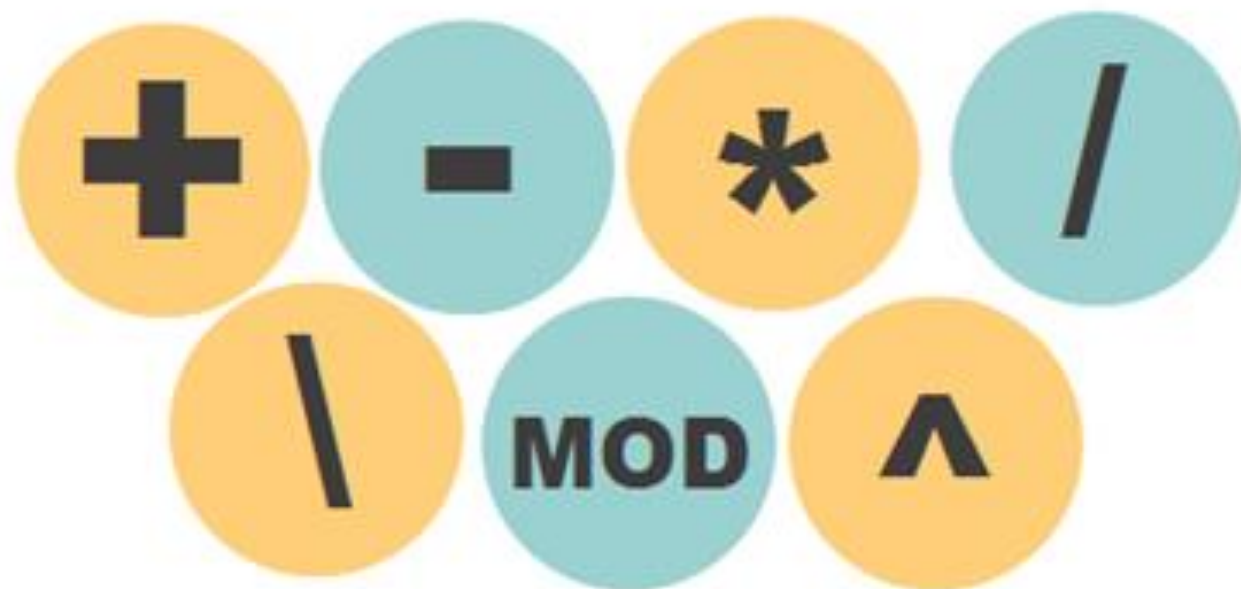
- Namespace/Packages are named: com.domainName.module name
- Namespace/Packages use only small letters
- Methods naming: methods start with **small** letter and Subsequent words start with **capital letters**
- Classes naming: Starts with **capital** and Subsequent words start with **capital letters**

# Agenda

- About Course
- Automation reminder
- Automation types
- Basic concepts of web applications
- What is Website?
- DOM
- HTML, CSS, JS - Basics
- Why Java?
- Java variables
- **Arithmetic Operators**
- Practice

Today, lets learn....

# Arithmetic Operators



They help you to perform basic operations on your data. Lets see how....?

# Arithmetic operators overview

Operator	Operation
+	Addition
-	Subtraction
/	Division
*	Multiplication
%	Remainder

# Operators breakdown: addition

## Integer numbers

```
int a = 10;  
int b = 30;  
int result = a + b;
```

---

*//result is 40*

## Fractional numbers

```
double x = 1.5;  
double y = 2.7;  
double result = x + y;
```

---

*//result is 4.2*



# Operators breakdown: subtraction

## Integer numbers

```
int a = 30;  
int b = 20;  
int result = a - b;
```

---

*//result is 40*

## Fractional numbers

```
double x = 5.4;  
double y = 1.6;  
double result = x - y;
```

---

*//result is 3.8*

# Operators breakdown: multiplication

## Integer numbers

```
int a = 2;  
int b = 4;  
int result = a * b;
```

---

*//result is 8*

## Fractional numbers

```
double x = 2.5;  
double y = 6.4;  
double result = x * y;
```

---

*//result is 16*

# Operators breakdown: division

## Integer numbers

```
int a = 10;  
int b = 5;  
int result = a / b;
```

---

*//result is 2*

## Fractional numbers

```
double x = 18.0;  
double y = 4.8;  
double result = x / y;
```

---

*//result is 3.75*

# Operators breakdown: remainder

## Integer numbers

```
int a = 9;  
int b = 6;  
int result = a % b;
```

---

*//result is 3*

## Fractional numbers

```
double x = 10.0;  
double y = 4.5;  
double result = x % y;
```

---

*//result is 1.0*

# Tricky questions

Type for division result is integer?

```
int a = 10;  
int b = 4;  
int result = a / b;
```

---

*//result is ?*

Type for division result is double?

```
int x = 10;  
int y = 4;  
double result = x / y;
```

---

*//result is ?*

# Type conversion: casting

- ▶ Operations with **widening** result require explicit type conversion (cast)

```
int x = 10;  
int y = 4;  
double result = x / (double) y;
```

---

*//result is ?*

# Writing output to the console: syntax

- ▶ Write to the console **content** of the *"args"* variable

```
System.out.println(args);
```

- ▶ Write to the console **directly** without variable

```
System.out.println("Hello World");
```

# Writing output to the console: example

## Printing Variable

```
int a = 10;  
String b = "Hi";  
System.out.println(a);  
System.out.println(b);
```

---

10

Hi

Process finished with exit code 0

## Printing directly

```
System.out.println(374);  
System.out.println("A");
```

---

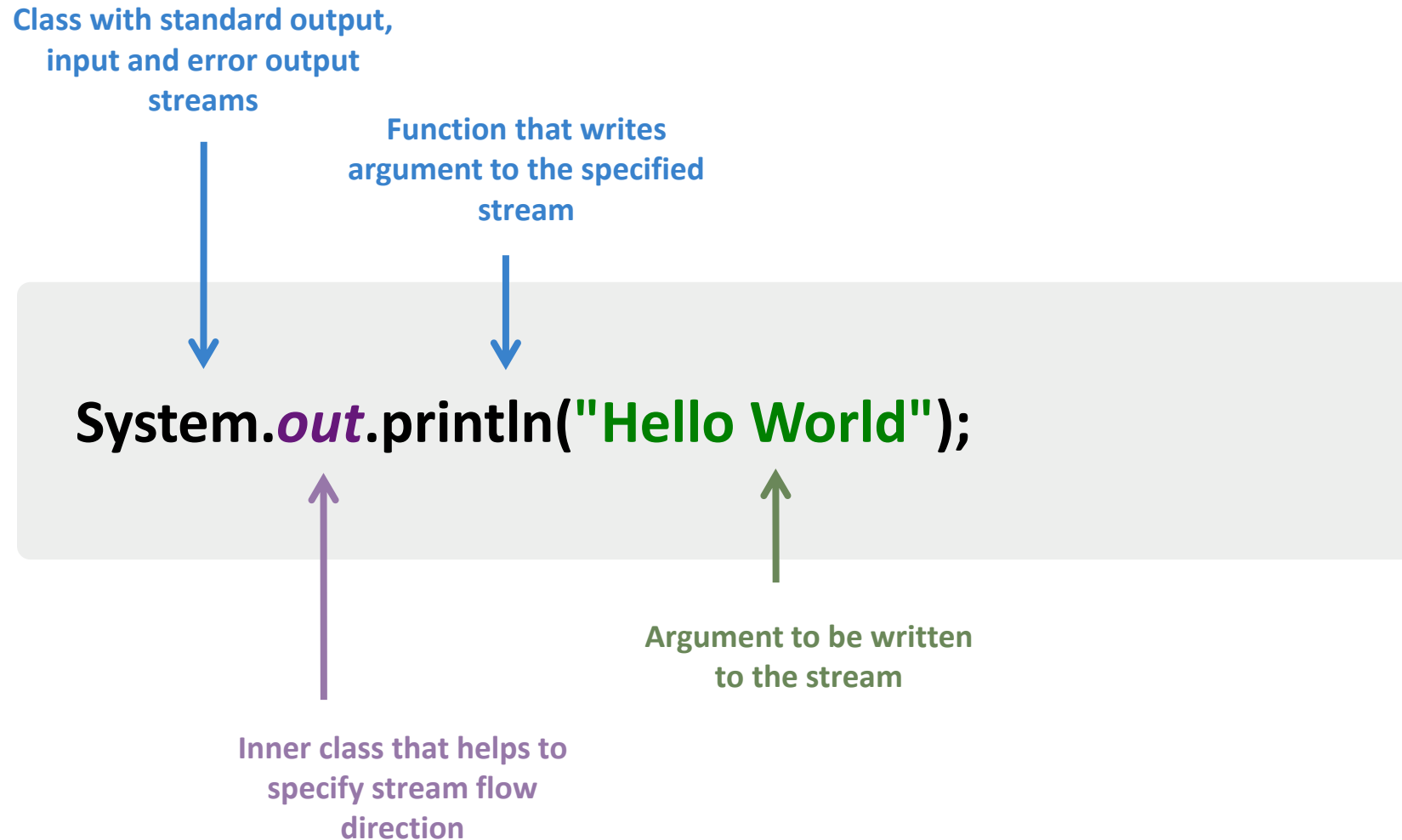
374

A

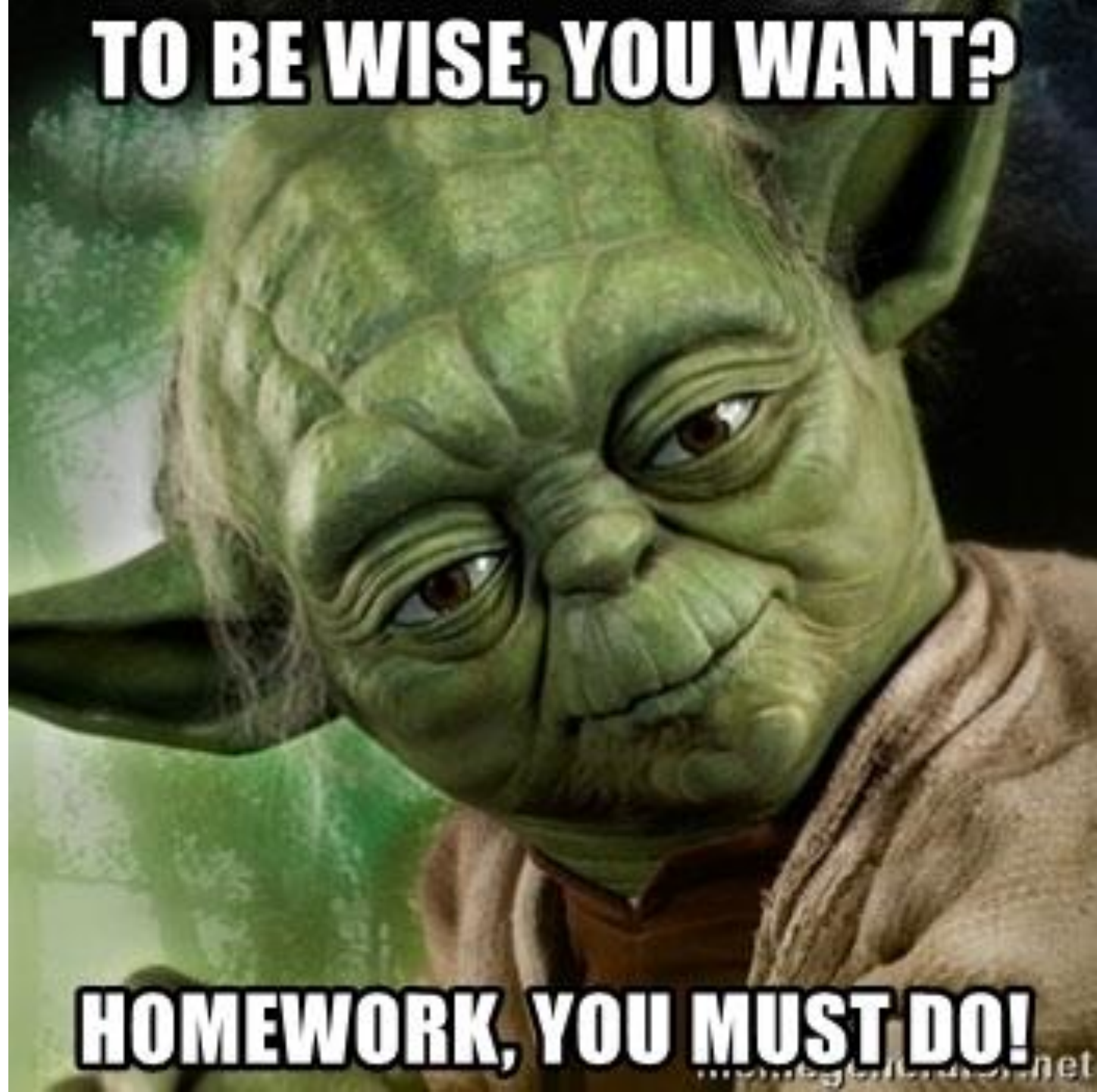
Process finished with exit code 0



# Writing output statement breakdown



**TO BE WISE, YOU WANT?**



**HOMEWORK, YOU MUST DO!**

# Agenda

- About Course
- Automation reminder
- Automation types
- Basic concepts of web applications
- What is Website?
- DOM
- HTML, CSS, JS - Basics
- Why Java?
- Java variables
- Arithmetic Operators
- **Practice**

# HTML Warm up

- Step 1: Open <https://codepen.io/AJamesL/pen/MXmvZp>
- For HTML PART
- Step 2: Change h1 text to Borrowing
- Step 3: Add to list word Borrowing
- Step 4: Change h2 text to Something
- Step 5: Change first paragraph to: “Scooby doo, Where are you?”
- Step 6: Leave this opened until further need

# CSS Warm up

- Step 1: Open our <https://codepen.io/AJamesL/pen/MXmvZp>
- Step 2: Lets change some colors
- Step 3: Change body background color to yellow
- Step 4: Change font color to blue

# JavaScript Warm up

- Step 1: Open <https://codepen.io/cphemm/pen/reNwWd>
- Step 2: Change html values so no matter what you have chosen in dropdown how was your service, it all the time calculated 30 percent
- Step 3: Multiply final tip sum by 100

# Warm up – Type and Name

- ▶ 1. Ship count in harbor-?
- ▶ 2. integer -?
- ▶ 3. first number-?
- ▶ 4. answer to the question?
- ▶ 6. random generated number-?
- ▶ 7. colour of the car-?
- ▶ 8. vehicle speed -?
- ▶ 9. user enter number to the console -?
- ▶ 10. biggest number ?
- ▶ 11. smallest number ?
- ▶ 15. question answer option -?
- ▶ 16. bank account-?
- ▶ 17. account balance -?
- ▶ 18. bank card-?
- ▶ 19. bicycle wheel number-?

# Lets setup everything

- <https://www.jetbrains.com/idea/download/#section=windows>



*That's all Folks!*



# Reference

- <https://www.w3schools.com/html/>
- <https://www.w3schools.com/css/>
- <https://docs.oracle.com/javase/tutorial/java/nutsandbolts/datatypes.html>
- <http://tutorials.jenkov.com/java/variables.html>
- <http://tutorials.jenkov.com/java/data-types.html>
- <https://javapapers.com/core-java/system-out-println/>



**THANK YOU FOR YOUR ATTENTION**

