

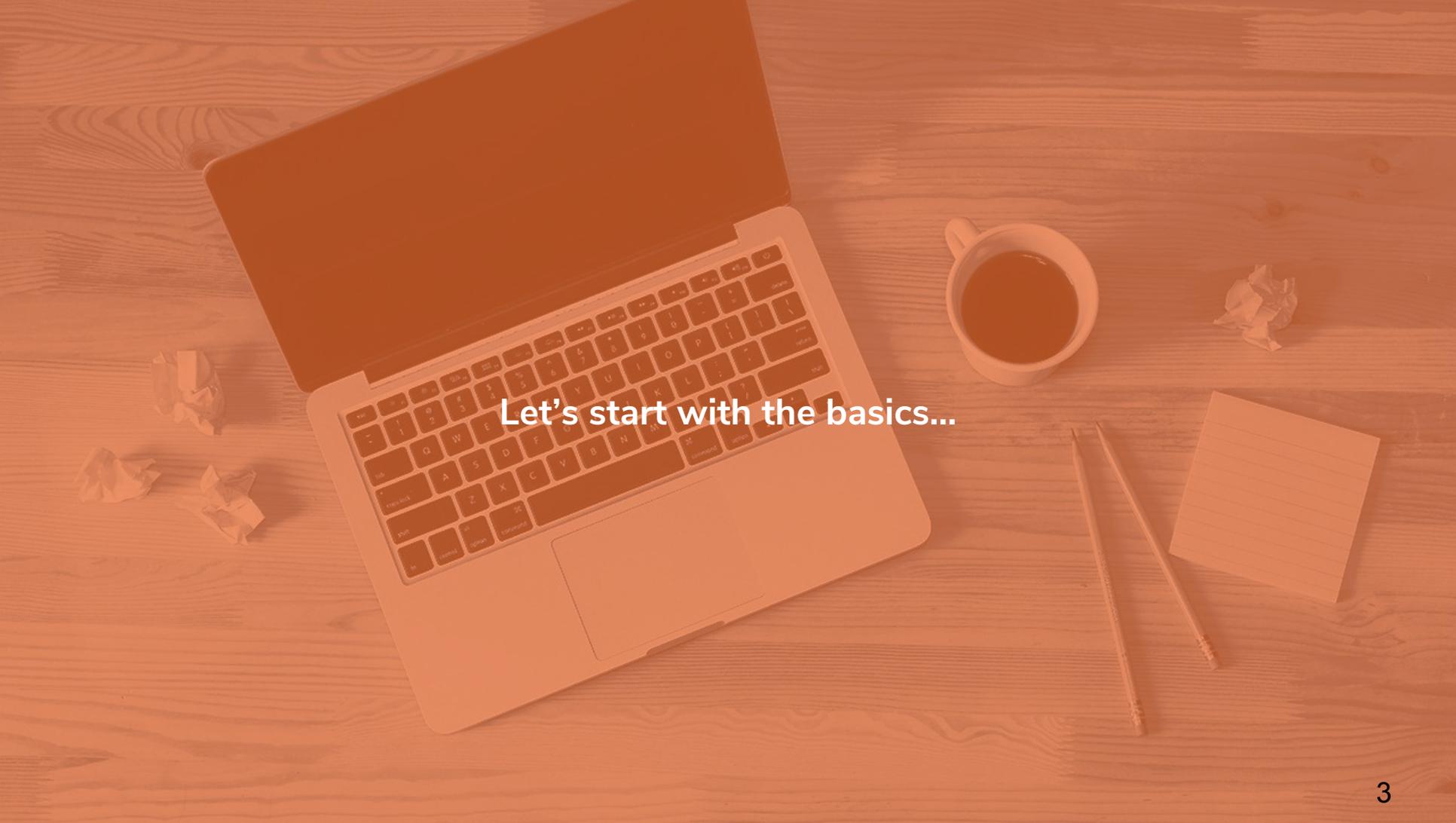
Introduction to the Basics of Selenium



We would be grateful if you could use Wi-Fi only for the workshop purposes and turn off all unrelated apps, updates etc., so as we could prevent ourselves from Wi-Fi lag.



First Thing First

A top-down view of a wooden desk with a laptop, a cup of coffee, pencils, and crumpled paper. The scene is bathed in a warm, orange light. The laptop is open, showing its keyboard and trackpad. To its right is a white mug filled with coffee. Below the mug are two pencils and a small, lined notepad. To the left of the laptop are several pieces of crumpled paper.

Let's start with the basics...

HTML

- Html is a standard markup language for creating Web pages. Html elements are the building blocks of HTML pages. HTML tags label pieces of content, such as “heading”, “paragraph”, “table”, and so on.
- HTML elements usually consists of start tag and end tag with content inserted between them.
- For example:
 - `<h1>` An example for an HTML title `</h1>`
 - `<p>` An example for an HTML paragraph `</p>`

HTML

```
<!DOCTYPE html>  
<head>  
<title>Page Title</title>  
</head>  
<body>  
  
<h1>This is part of the presentation title</h1>  
<p>This is the paragraph.</p>  
  
</body>  
</html>
```

CSS

- CSS is a language that describes the style of an HTML document.
- CSS describes how HTML elements should be displayed.
- A CSS rule-set consists of a selector and a declaration block. For example: selector - h1, declaration – {color:blue;}.

▪An Example:

```
<style>  
body {  
  background-color: lightblue;  
}  
</style>
```

The body of the HTML page will be with a light blue background.

XPATH

- *XPath stands for XML Path Language, it can be used to navigate through elements and attributes in an XML document. XPath uses path expressions to select nodes or node-sets in an XML document.*

XPath stands for XML Path Language, it can be used to navigate through elements and attributes in an XML document. XPath uses path expressions to select nodes or node-sets in an XML document.

Syntax:

- // - Selects nodes in the document from the current node that match the selection no matter where they are.
- @ - Selects attributes

XPATH

- *Example*

- ```
<!DOCTYPE html>
<html>
<head>
<title>Page Title</title>
</head>
<body>
<h1 id="h1_id">This is part of the presentation title</h1>
<p>This is the paragraph.</p>
</body>
</html>
```

The query: `//h1[@id='h1_id']` will get the h1 element

# CSS Selector

- CSS selectors are used to "find" (or select) HTML elements based on their element name, id, class, attribute, and more.
- An example for some of the selectors are:
  - Element selector : The element selector selects elements based on the element name. for example: p, h1 etc.
  - ID selector: The id selector uses the id attribute of an HTML element to select a specific element. For example: #id1, #para2 etc.

# CSS Selector

- A CSS rule-set consists of a selector and a declaration block.  
For example: selector - `h1`, declaration – `{color:blue;}`.

# Java in a Nutshell

- Java technology is used to develop applications for a wide range of environments, from consumer devices to heterogeneous enterprise systems.
- Java is an object-oriented programming language with its own runtime environment.
- Java programs are platform independent which means they can be run on any operating system with any type of processor as long as the Java interpreter is available on that system.

# Java in a Nutshell

- **Example:**

```
public class Sample {

 public static void main(String args[]) {

 int a = 80, b = 30;

 if (b > a) {

 System.out.println("b is greater");

 } else {

 System.out.println("a is greater");

 }

 }

}
```

# Maven in a Nutshell

Let's keep it simple.

- Maven allows a project to build using its project object model (POM) and a set of plugins that are shared by all projects using Maven, providing a uniform build system.
- Using Selenium requires to use external dependencies, Maven allows to easily integrate Selenium to our project.

- For example:

Adding to the pom file in a Maven Java project the following dependency will give the ability to use Selenium as it is part of the project, meaning as if the jars are already in the project.



# Selenium

Automating browser interaction got a lot easier

# Selenium

*Selenium is used to automate browsers. It is a collection of tools used across many platforms. There are several parts of Selenium, we will focus on Selenium WebDriver.*

- The Selenium WebDriver is a collection of language specific binding, used to automate the browser.
- In the Automation, Selenium WebDriver is used in a Java Maven project. It enables us to automate interaction with anything needed in the web.
- Most common supported browsers by Selenium WebDriver are: Chrome, Internet explorer and Firefox.

## *Creating a new Selenium WebDriver*

- A Selenium Web driver in FireFox can be created over one line of code. Other browsers, such as Internet and Chrome are supported, as well.
- For Example:
  - `WebDriver driver = new FirefoxDriver();`
- Interaction with the Firefox instance will be made in the code on the driver.

# Selenium

## *Creating a new Selenium WebDriver*

- Some companies would like to test their product in more than one browser (depends on the product).
- It is important to think how the Selenium driver can be handled in a generic matter.
- Meaning we will like our test to be executed using the same code on more than one browser.

## *Navigating Using a Selenium WebDriver*

- Navigation using a Selenium WebDriver is very simple, given a defined URL. It can be done in two ways, `driver.get(...)` or `driver.navigate().to(...)`.
- For example:
  - `driver.get("https://www.google.com/");`
  - `driver.navigate().to("https://www.google.com/");`
- The `driver.get(...)` and `driver.navigate().to(...)` do exactly the same thing. `driver.navigate()` supports also `driver.navigate().forward()` and `driver.navigate().backward()` .

## *Web Elements and How to Find Them*

- Web elements can be defined as each opening and closing tags in the web page. For example: `<button>Click Me</button>` - a web element.
- Finding a web element and interacting with it can be done in several ways:
  - ID.
  - Class.
  - Name.
  - Xpath.
  - Css Selector, etc.

## *Web Elements and How to Find Them*

- An example:

- Assuming that we have the following web page:

```
<html>
 <body>
 <button id="my_button">Click Me</button>
 </body>
</html>
```

- The following lines of code will be used for clicking the button:

```
WebElement button = driver.findElement(By.id("
my_button "));
button.click();
```

# Exercise

## *Navigate to Google and search exercise:*

- Create a new Java Maven Project.
- Import Selenium to the project using .pom file.
- Open Chrome browser.
- Navigate to google.com .
- search the following value:
  - Selenium

# Exercise

## *Troubleshooting:*

- Error: Exception in thread "main"  
java.lang.UnsupportedClassVersionError:  
org/openqa/selenium/WebDriver : Unsupported major.minor  
version 52.0
- Solution: need to use JDK 1.8 or use lower version of Selenium, for  
example: version 2.53.1.

## Links

*Extra information or self learning can be achieved at the following links:*

- [Selenium site](#)
- [Selenium Tutorial](#)
- [W3Schools site](#)



## Thank you very much for your time

If you have any questions about this document  
please don't hesitate to contact:

- [saarr@gett.com](mailto:saarr@gett.com)
- [@Gett](#)

