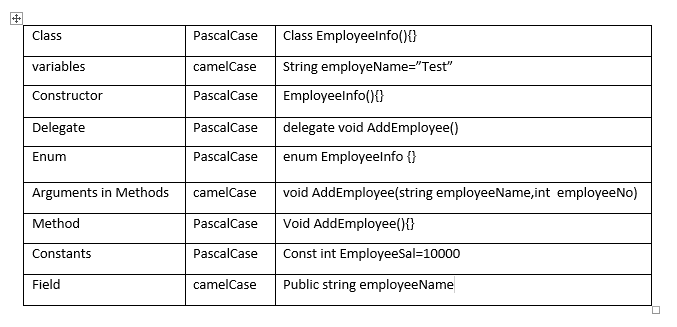
**LINKS**

**Good C# practices:**

* <https://vaibhavbhapkarblogs.medium.com/c-best-practices-48e984cfdac0>



**HTML element references:**

1. <https://developer.mozilla.org/en-US/docs/Web/HTML/Element>

The **<html>** [HTML](https://developer.mozilla.org/en-US/docs/Web/HTML) element represents the root (top-level element) of an HTML document, so it is also referred to as the *root element*. All other elements must be descendants of this element.

**HTML** – HyperText Markup Language. HyperText refers to the fact that HTML makes that you can click on links on websites. Markup Language – something you use to markup normal English into Hypertext/to indicate things.

**Hypertext** is the medium used to transmit the information in a non-linear fashion via computer by clicking on a "link" using a mouse.

HTML describes the purpose of the text. Then, it’s up to the web browser to know what to do with that text.

HTML usually comes in two tags – the opening and the closing (e.g. <strong>TITLE</strong> (bold font)).

<h1 class = “article”> (h1 – tag, class – attribute, “article” – value). **The document written in HTML defines the structure and content of the page.**

HTML describes the text that it is marking up.

**Tags examples:**

Headers (indicate hierarchy to text): <h3>Subject</h3> (3 is the size).

Table (<table>) - creates a table for items (like the one in above picture in this word file).

Source (src=https://www…) = sources for images.

reference: <https://www.youtube.com/watch?v=u0OeZfIfBRI&ab_channel=WinningWP-WinningWordPress>

what is hypertext: <http://mason.gmu.edu/~montecin/hypertext.htm>

**WEB development** – is made of HTML file, CSS (Cascading Style Sheets) file and JavaScript file.

CSS is responsible for style. (How things look on the website, the positioning, colors, simple animation, etc).

To tie structure and style together, the HTML document can provide the browser with information on what style to use and where to get it.

(e.g. css can command <h1 class = “article” to be in blue>).

**CSS Cascades through 3 manners:**

* Elements (color : blue)
* Multiple styles – One element (font color : yellow)
* Sheets – multiple styling written, which can be applied to multiple websites to achieve unique designs (the design will differ from each and every website according to how things are styled).

**CSS selectors:**

Selecting elements in HTML page.

* <div class = “button”> - can be added to several elements/multiple classes to one element.

. functionality (**.button** { font – color : yellow;}…)

* <div id = “submit”> - to identify a very unique element on a HTML page. Cannot be shared.

# functionality (**#submit** {position : absolute;}…)

**CSS <property> & <value>:**

* font – color : yellow; (font – color – property, yellow – value).
* Multiple types of properties to enhance the design:

border: 1px solid blue

reference: <https://www.youtube.com/watch?v=6vbgZnQrpbU&ab_channel=edureka%21>

**CSS selectors types:**

* ID
* Class
* Attribute
* Sub-String
* Inner String

The broad steps **to perform a test** through Selenium are as follows:

1. Download browser drivers
2. Initiate a Selenium WebDriver
3. Load web applications
4. Perform designated actions in a defined test
5. Assess if the test achieved the desired outcome
6. Close the WebDriver

Locators in Selenium come into action in the fourth step above, after the Selenium WebDriver is initialized and loaded the webpage to be tested. A locator enables testers to select an HTML DOM element to act on.

**The different locators in Selenium are as follows:**

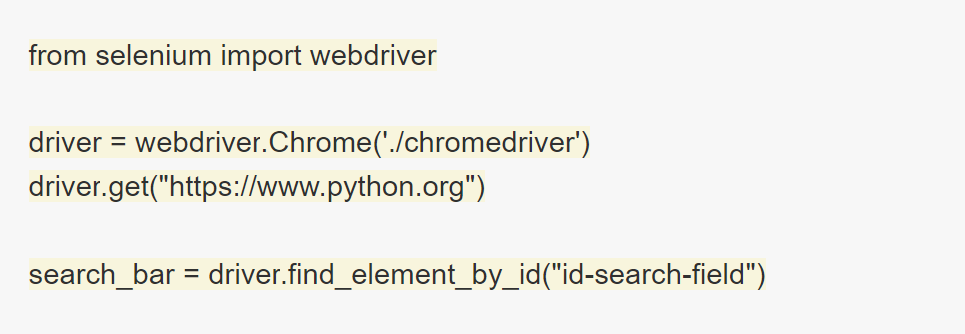
* By CSS ID: *find\_element\_by\_id*
* By CSS class name: *find\_element\_by\_class\_name*
* By name attribute: *find\_element\_by\_name*
* By DOM structure or xpath: *find\_element\_by\_xpath*
* By link text: *find\_element\_by\_link\_text*
* By partial link text: *find\_element\_by\_partial\_link\_text*
* By HTML tag name: *find\_element\_by\_tag\_name*

While all these locators return single elements, one may use the *.find\_elements()* method to find multiple elements.

**Locate Elements by CSS ID**

This is by far the simplest method of locating an element. The CSS ID, stored in the *id* attribute of an **HTML DOM** (**Document Object Model)** element, is unique for every element in the page by design. Thus, an ID can uniquely identify an element.

To use this feature, one needs to call the *.find\_element\_by\_id()* method of the *webdriver* class. Here is the usage for it.

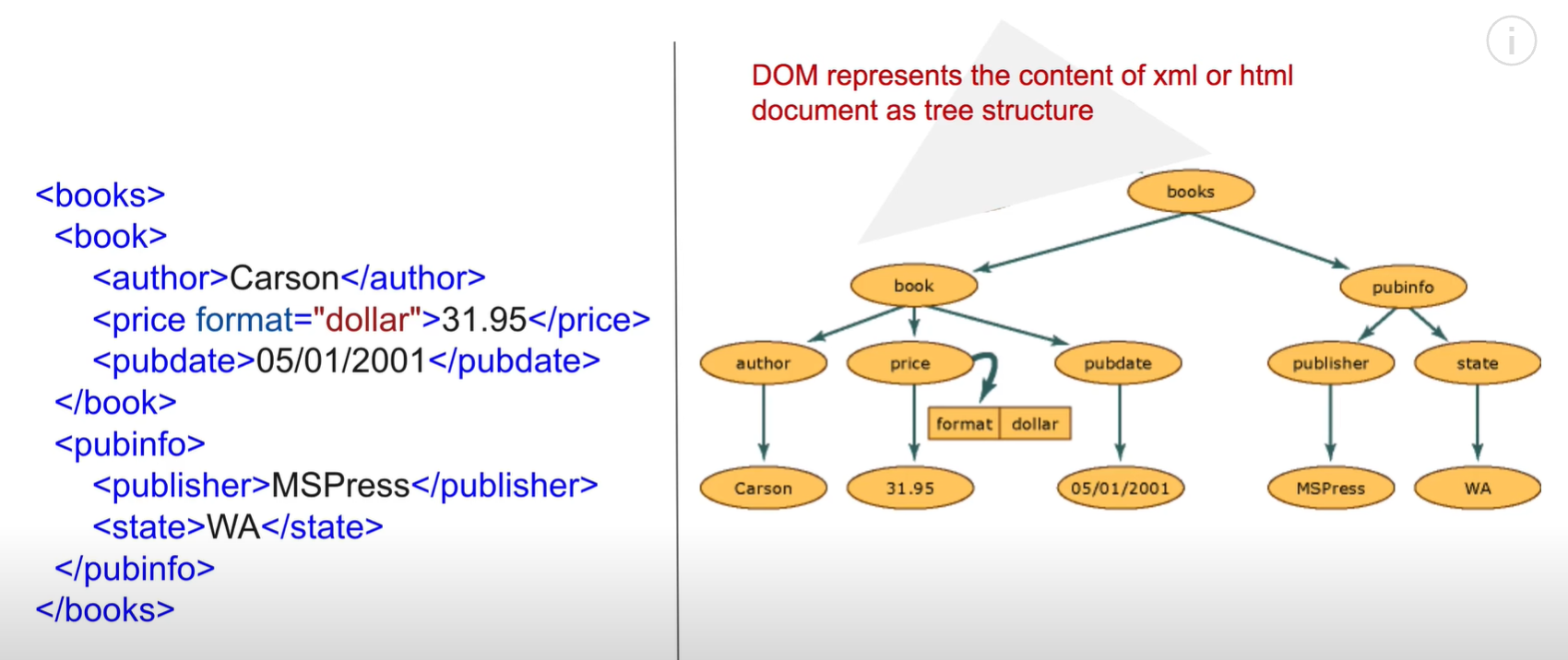


If there is no DOM element with the ID that one is searching for, a *NoSuchElementException* is raised, which one can account for, by using a try-catch block.

Theoretically, every DOM element on a page should have a unique ID. Most elements may not have an ID or may encounter two elements with the same ID. In such cases, one needs to use a different strategy to identify a DOM element uniquely.

**HTML DOM** represents the content of xml or html document as tree structure.

The elements of **DOM** is head,title,body tag etc.



**DOM** is mostly used in JavaScript, but can be used in any language,

Using DOM, we can easily read, access, and update the contents of the document.

**DOM** makes the tree structure internally in the memory, and then using the program of the language that we are using we can then use it to update and access the contents.

**DOM (Document Object Model):**

* Document – the file
* Object – tags, elements
* Model – layout, structure

Is an API (programming interface)

All XML elements can be accessed through the XML DOM:



Using the inspect tool on a specific button on a website, we can access the elements and their locators. When we do automation of web browsers using tools like Selenium, we have to use the inspect tool to access the elements using XPath or any other locators. DOM is being used in the back end when we change the elements or do any modifications.