Structure of HTML 5

Bok, Jong Soon
javaexpert@nate.com
https://github.com/swacademy/HTML5

HTML5 Syntax Checks

- Uppercase, lowercase, camel case
- Attributes quoted, attributes not quoted
- Self-closing or not
- No deprecated elements, only obsolete elements – don't use them, but they work

HTML Element Syntax

- HTML documents consist of a tree of ***element***s and ***text***.
- Each ***element*** is denoted by a start tag / opening tag, such as **<body>**, and an end tag / closing tag, such as **</body>**.

```
<p>This is <em>very <strong>wrong</em>!</strong></p>

<p>This <em>is <strong>correct</strong>.</em></p>
```

- The ***element*** content is everything between the start and the end tag.
- Some HTML ***element***s have empty content.
- Empty ***element***s are closed in the start tag.

HTML Attribute Syntax

- ***Element***s can have ***attribute***s, which control how the ***element***s work.

```
<a href="demo.html">simple</a>
```

- ***Attribute***s are placed inside the start tag, and consist of a ***name*** and a ***value***, separated by an = character.
- The ***attribute*** value can be left unquoted if it is a keyword, but generally will be quoted.
- The ***value*** can also be omitted altogether if it is empty.

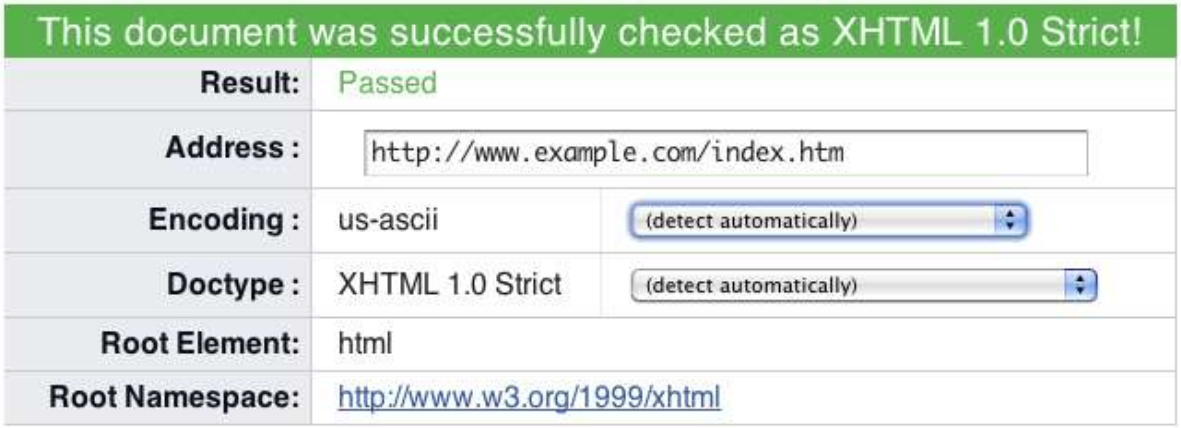
HTML Comment

- The comment tag is used to insert comments in the source code.
- Comments are not displayed in the browsers.
- You can use comments to explain your code, which can help you when you edit the source code at a later date.
- <!-- -->

```
1  <!--This is a comment. Comments are not displayed in the browser-->
2
3  <p>This is a paragraph.</p>
```

Valid HTML vs. Valid XHTML

- So you created XHTML…



- But did you know that
  - Over 90% of XHTML is delivered with the **`text/html`** MIME type and therefore broken.
  - Delivering XHTML as **`text/xml`** or **`application/xhtml+xml`** is risky(not supported in old IE).

Valid HTML vs. Valid XHTML (Cont.)

- Use HTML5 and create valid HTML
  - http://validator.w3.org
  - http://validator.nu/
- HTML5 allows XML syntax from XHTML 1.0 for backward compatibility
  - `<br />`
- HTML5 has well-defined processing rules
- Simple is better
  - New *doctype*
  - Character set

<!DOCTYPE>

- Must be the very first thing in your HTML document, before the `<html>` tag.
- Is not an HTML tag; it is an instruction to the web browser about what version of HTML the page is written in.
- In HTML 4.01, the `<!DOCTYPE>` declaration refers to a DTD, because HTML 4.01 was based on SGML.
- The DTD specifies the rules for the markup language, so that the browsers render the content correctly.
- HTML5 is not based on SGML, and therefore does not require a reference to a DTD.

<!DOCTYPE> (Cont.)

- Does not have an end tag.
- Is ***NOT*** case sensitive.
- Use W3C's Validator to check that you have written a valid HTML / XHTML document!

<!DOCTYPE> (Cont.)

- HTML 4.01

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
```

- XHTML 1.0

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

- HTML 5

```
<!doctype html>
```

Simplified Doctype

HTML4/XHTML

- HTML 4.01 Strict
- HTML 4.01 Transitional
- HTML 4.01 Frameset
- XHTML 1.0 Strict
- XHTML 1.0 Transitional
- XHTML 1.0 Frameset
- XHTML 1.1

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML
4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
```

HTML5

```
<!DOCTYPE html>
```

VS Code 에서 ! 탭을 하면 html 기본 양식을 자동완성해준다.

HTML 각 Element를 확인하기 위한 API 문서
https://html.spec.whatwg.org/multipage/

HTML 테스트, 샘플 등 학습자료를 제공하는 페이지
https://www.w3schools.com/html/default.asp

평생직업능력 개발원 인강 https://www.step.or.kr/

<html>

- Tells the browser that this is an HTML document.
- Represents the root of an HTML document.
- Is the container for all other HTML elements (except for th e `<!DOCTYPE>` tag).

<html> (Cont.)

- Differences between HTML 4.01 and HTML5.
  - HTML5 has added a new attribute: **`manifest`**.
  - **`manifest=URL`**
  - Specifies the address of the document's cache manifest (for offline browsing)
- Differences Between HTML and XHTML
  - The **`xmlns`** attribute is required in XHTML, but is invalid in HTML.
  - **`xmlns=http://www.w3.org/1999/xhtml`** is default.

<html> lang Attribute

- Sets the language code.

```
1  <!DOCTYPE html>
2  <html lang="ko">
3      <head>
4          <title> New Document </title>
5          <meta charset="utf-8">
6      </head>
7      <body>
```

Korean	English	Chinese	French	Arab	Japanese	Spanish
ko	en	zh	fr	ar	ja	es

Language code ref : http://rishida.net/utils/subtags/

manifest는 오프라인 용 Attribute (캐싱)
-> 안 다룸

<html> manifest Attribute

- Specifies the location of the document's cache manifest.
- HTML5 introduces application cache.
  - Means that a web application is cached, and accessible without an internet connection.
  - Gives an application three advantages:
    - Offline browsing - users can use the application when they're offline.
    - Speed - cached resources load faster.
    - Reduced server load - the browser will only download updated/changed resources from the server.

<html> manifest Attribute (Cont.)

- Should be included on every page of your web application that you want cached.
- The manifest file is a simple text file that lists the resources the browser should cache for offline access.

```
1  <!DOCTYPE HTML>
2  <html manifest="demo.appcache">
3      <head>
4          <title>Title of the document</title>
5      </head>
6
7      <body>
8          The content of the document.....
9      </body>
10 </html>
```

<html> `manifest` Attribute (Cont.)

- Syntax
  - `<html manifest="URL">`
- URL
  - The address of the document's cache manifest.
  - Possible values:
    - An absolute URL - points to another web site (like href="http://www.example.com/demo.appcache")
    - A relative URL - points to a file within a web site (like href="demo.appcache")
- Internet Explorer 10+, Firefox, Chrome, Safari and Opera support Application cache.

브라우저는 태그를 이용해 렌더링한다.
<head>는 다음과 같은 태그들을 포함하는 Element
<title>, <style>, <base>, <link>, <meta>, <script>, <noscript>
-> 명시하지 않아도 페이지는 동작하지만, 문법상 오류라 반드시 포함하는 태그들이다.

<head>

- Is a container for all the head elements.
- Must include a title for the document, and can include scripts, styles, meta information, and more.
- The following elements can go inside the **<head>** element:
  - **<title>, <style>, <base>, <link>, <meta>, <script>, <noscript>**
- Differences Between HTML 4.01 and HTML5
  - The **profile** attribute is not supported in HTML5.

<title>

- Is required in all HTML documents and it defines the title of the document.
- Defines a title in the browser toolbar.
- Provides a title for the page when it is added to favorites.
- Displays a title for the page in search-engine results.
- Can *NOT* have more than one `<title>` element in an HTML document.
- If you omit the `<title>` tag, the document will not validate as HTML.

브라우저 상단 탭에 출력되는 페이지 명

<style>

- Is used to define style information for an HTML document.
- Inside, specify how HTML elements should render in a browser.
- Each HTML document can contain multiple **<style>** tags.

```
<!-- 1. 인라인으로 태그 별로 스타일을 적용할 수 있다. 하지만 일반적으로 사용 X -->
 <h1 style="background-color:black, color= white, text-align= center">
Hello, World
</h1>
```

```
<!-- 2. Head에 Style을 미리 정의한다.
  body의 태그 중 h2을 만나면 h2에 Attribute를 적용함.
  MVC 패턴과 마찬가지로 일반적으로는 HTML 내에서 스타일 선언 X -->
<head>
  <style type="text/css">
    h1 {
        background-color: black;
        color: yellow;
        text-align: center;
    }
  </style>
</head>
```

```
 8    <body>
 9        <h1>A heading</h1>
10        <p>A paragraph.</p>
11    </body>
12  </html>
```

<style> (Cont.)

```
<!-- 3. 외부 경로의 css 파일을 불러와 스타일을 적용.
  가장 일반적인 방법으로 이 경우에는 로컬 파일 뿐만 아니라
글로벌 서버에 등록된 CSS 파일을 불러올 수 있다. (Ex. 부트스트랩)
  선언하는 위치는 <head>  이다.  -->
 <link rel="stylesheet" href="./style.css">
```

- To link to an external style sheet, use the **`<link>`** tag.
- If the **`scoped`** attribute is not used, each **`<style>`** tag must be located in the head section.
- Differences Between HTML 4.01 and HTML5
  - The **`scoped`** attribute is new in HTML5.

<style> (Cont.)

- Attributes
  - **media**
    - Specifies what media/device the media resource is optimized for.
  - **scoped** 

    - Specifies that the styles only apply to this element's parent element and that element's child elements
  - **type**
    - Specifies the MIME type of the style sheet.

MIME는 어떤 파일인지 나타내는 식별자.
Major 분류 / Minor 분류로 구분된다.
ex)
text/html, text/css, text/javascript 등
text 문서 -> javascript 문서

<style> media Attribute

- Specifies what media/device the CSS style is optimized for.
- Is used to specify that the style is for special devices (like iPhone), speech or print media.
- Can accept several values.

```
1  <style type="text/css" media="print">
2      h1 {color:#000000;}
3      p {color:#000000;}
4      body {background-color:#FFFFFF;}
```

<style> `media` Attribute (Cont.)

- Possible Operators : `and`, `not`, `,`
- Devices
  - all : default, suitable for all devices.
  - aural : Speech synthesizers
  - braille : Braille feedback devices.
  - handheld : Handheld devices
  - projection : Projectors
  - print : Print preview mode / printed pages
  - screen : Computer screens
  - tty : Teletypes and similar media using a fixed-pitch character grid.
  - tv : Television type devices

<style> scoped Attribute

- Is currently supported only in Firefox.
- Is a boolean attribute.
- Syntax
  - `<style scoped>`

```
1  <div>
2      <style type="text/css" scoped>
3          h1 {color:red;}
4          p {color:blue;}
5      </style>
6      <h1>This is a heading</h1>
7      <p>This is a paragraph.</p>
8  </div>
```

- When present, it specifies that the styles only apply to this element's parent element and that element's child elements (not the entire document).

<base>

- Specifies the base URL/target for all relative URLs in a document.
- There can be at maximum one **<base>** element in a document, and it must be inside the **<head>** element.

```
1  <head>
2      <base href="http://www.w3schools.com/images/" target="_blank">
3  </head>
4
5  <body>
6      <img src="stickman.gif" width="24" height="39" alt="Stickman">
7      <a href="http://www.w3schools.com">W3Schools</a>
8  </body>
```

<base> (Cont.)

- Put the **<base>** tag as the *first* element inside the **<head>** element, so that other elements in the head section uses the information from the **<base>** element.
- If the **<base>** tag is present, it must have either an **href** attribute or a **target** attribute, or both.
- In HTML the **<base>** tag has no end tag.

<base> (Cont.)

- Attributes
  - **`href`**
    - Specifies the base URL for all relative URLs in the page.
  - **`target`**
    - Specifies the default target for all hyperlinks and forms in the page.
    - **`_blank`**, **`_parent`**, **`_self`**, **`_top`**, framename

<link>

- Defines the relationship between a document and an external resource.
- Is most used to link to style sheets.
- This element goes only in the head section, but it can appear any number of times.

```
1  <head>
2      <link rel="stylesheet" type="text/css" href="theme.css">
3  </head>
```

- Has no end tag.

<link> (Cont.)

- Attributes
  - **`href`**
    - Specifies the location of the linked document.
  - **`rel`**
    - Required.
    - Specifies the relationship between the current document and the linked document.
    - alternate, archives, author, bookmark, external, first, help, icon, last, license, next, nofollow, noreferrer, pingback, prefetch, prev, search, sidebar, stylesheet, tag, up

<link> (Cont.)

- Attributes
  - **`sizes`** 
    - Specifies the size of the linked resource.
    - Only for **`rel="icon"`**
  - **`type`**
    - Specifies the MIME type of the linked document.

<link> sizes Attribute

- Is not currently supported in any of the major browsers.
- Specifies the sizes of icons for visual media.
- This attribute is only used if **rel="icon"**.
- Syntax
  - **<link sizes="HeightxWidth|any>**

```
1  <link rel="icon" href="demo_icon.gif" type="image/gif" sizes="16x16">
```

<script>

- Is used to define a client-side script, such as a JavaScript.
- Either contains scripting statements, or it points to an external script file through the **`src`** attribute.
- Common uses for JavaScript are image manipulation, form validation, and dynamic changes of content.

```
1  <script>
2      document.write("Hello World!")
3  </script>
```

<script> (Cont.)

- Differences Between HTML 4.01 and HTML5
  - The `type` attribute is required in HTML 4, but optional in HTML5.
  - The `async` attribute is new in HTML5.
  - The HTML 4.01 attribute: `xml:space`, is not supported in HTML5.

<script> (Cont.)

- Attributes
  - **`async`**
    - Specifies that the script is executed asynchronously (only for external scripts)
  - **`charset`**
    - Specifies the character encoding used in an external script file.
  - **`defer`**
    - Specifies that the script is executed when the page has finished parsing (only for external scripts)

<script> `async` Attribute

- Is supported in Internet Explorer 10, Firefox, Opera, Chrome, and Safari.
- Is a boolean attribute.
- When present, it specifies that the script will be executed asynchronously as soon as it is available.
- Is only for external scripts (and should only be used if the `src` attribute is present).

```
1   <script src="demo_async.js" async></script>
```

<script> charset Attribute

- Specifies the character encoding used in an external script file.
- Is used when the character encoding in an external script file differs from the encoding in the HTML document.

```
1   <script src="myscripts.js" charset="UTF-8"></script>
```

<script> `defer` Attribute

- Is a boolean attribute.
- When present, it specifies that the script is executed when the page has finished parsing.
- Is only for external scripts (should only be used if the `src` attribute is present).

```
1   <script src="demo_defer.js" defer></script>
```

<meta>

- Provides metadata about the HTML document.
- Metadata will not be displayed on the page, but will be machine parseable.
- Are typically used to specify page description, keywords, author of the document, last modified, and other metadata.
- The metadata can be used by browsers (how to display content or reload page), search engines (keywords), or other web services.

<meta> (Cont.)

- Always goes inside the **<head>** element.
- Metadata is always passed as name/value pairs.
- Has no end tag.

```
1  <head>
2      <meta name="description" content="Free Web tutorials">
3      <meta name="keywords" content="HTML,CSS,XML,JavaScript">
4      <meta name="author" content="Ståle Refsnes">
5      <meta http-equiv="refresh" content="30">
6      <meta charset="UTF-8">
7  </head>
```

<meta> (Cont.)

- Attributes
  - **`charset`** 

    - Specifies the character encoding for the HTML document.
  - **`content`**
    - Gives the value associated with the **`http-equiv`** or **`name`** attribute.
  - **`http-equiv`** :
    - Provides an HTTP header for the information/value of the content attribute.
    - **`content-type`**, **`default-style`**, **`refresh`**

Character encoding ref : http://www.iana.org/assignments/character-sets/character-sets.xhtml

<meta> (Cont.)

- Attributes
  - **`name`**
    - Specifies a name for the metadata.
    - **`application-name`**, **`author`**, **`description`**, **`generator`**, **`keywords`**

<meta> (Cont.)

- <meta name="description" content="HTML5 연습">
- <meta name="author" content="Peter Bok">
- <meta name="copyright" content="Copyright © 2014 javaexpert.co.kr">
- <meta name="reply-to" content="javaexpert@nate.com">
- <meta name="date" content="2014-05-14T 12:00:00+09:00">

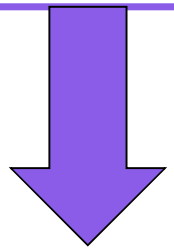
<meta> (Cont.)

- <meta http-equiv=“keywords” content=“HTML5, CSS, JavaScript”>
- <meta http-equiv =“Content-Type” content=“text/html;charset=euc-kr”>
- <meta http-equiv =“Pragma” content=“no-cache”>
- <meta http-equiv =“Refresh” content=“3;url=http://www.google.com”>

Simplified Character Set

HTML4

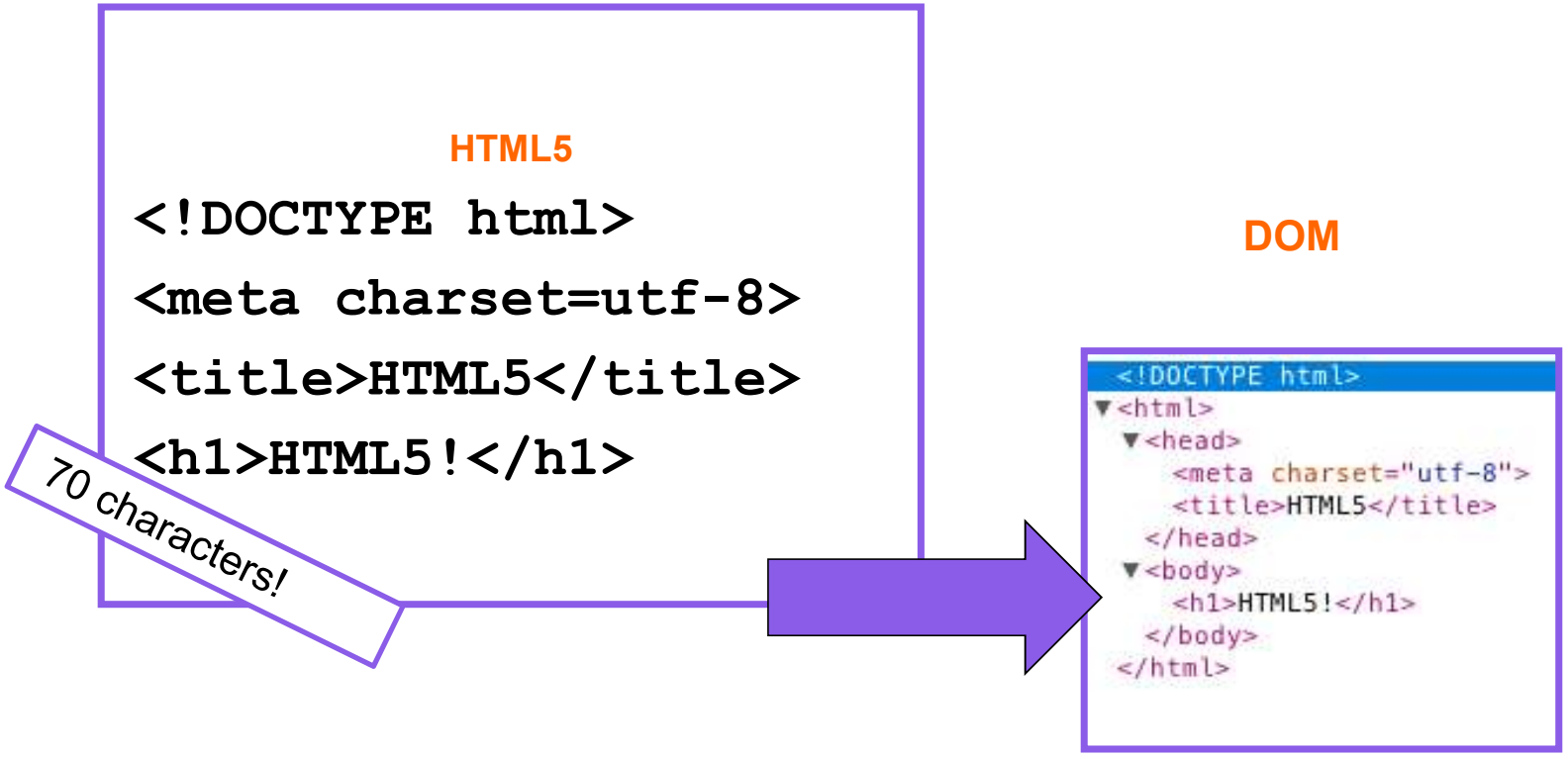
```
<meta http-equiv="Content-Type"
content="text/html; charset=utf-8">
```



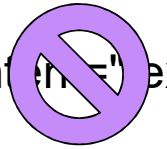
HTML5

```
<meta charset=utf-8>
```

Minimal HTML5 Page



Getting Started……



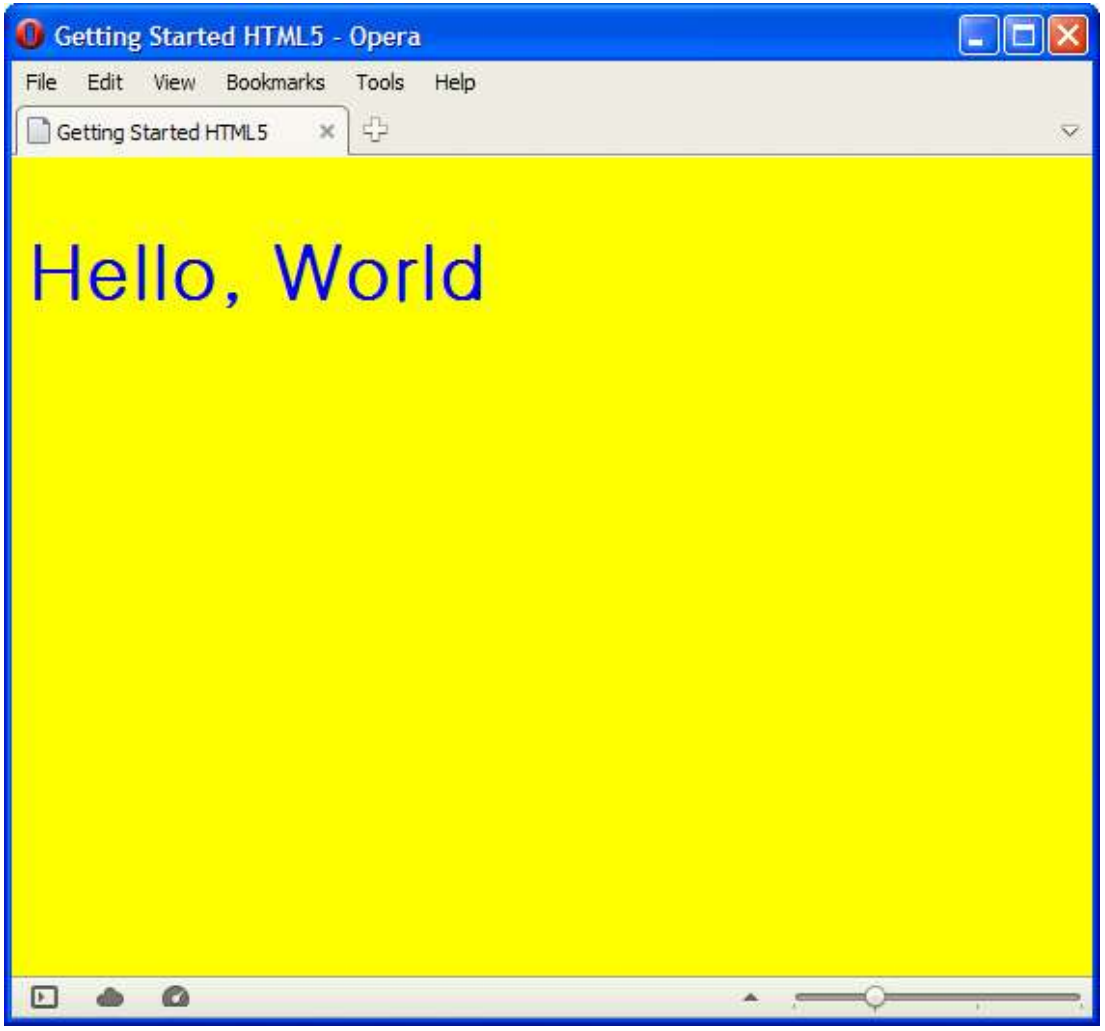
Lab1 : Hello, World

- Web Browsers
  - Edge, Firefox, Google Chrome, Opera, Safari
- Text Editors
  - Visual Studio Code, Notepad++, Editplus, etc…
- Files
  - helloworld.html

Lab1 : helloworld.html

```
1  <!doctype html>
2  <html>
3      <head>
4          <title>Getting Started HTML5</title>
5          <meta charset="utf-8">
6          <style type="text/css">
7              body { background-color :yellow }
8              p {font-size : 30pt; color : blue ; font-weight : 900 }
9          </style>
10     </head>
11     <body>
12         <p>Hello, World</p>
13     </body>
14 </html>
```

Lab1 : Result


Getting Started HTML5 - Opera
File Edit View Bookmarks Tools Help
Getting Started HTML5
Hello, World

Lab1 : http://validator.nu/