DCS 3003

WEBSITE DESIGN

HTML

Recap

HTML –

ActivityLesson 9

(Challenging)

CRESCENDO INTERNATIONAL COLLEGE
TIMETABLE FOR SEMESTER JANUARY 2023
DIPLOMA IN COMPUTER SCIENCE

Day/Time	Intake	9:00 - 11:00 AM	11:15 - 1:15 PM	2:15 - 4:15 PM
Monday	S6			
	S5			
	S4	DCS 4103	DCS 4103	DCS 4103
	S3			
	S2			
	S1			
Tuesday	S6			
	S5			
	S4			
	S3			
	S2	DCS 3003	MST 1003	MST 1003
	S1			
Wednesday	S6			
	S5			
	S4			
	S3			
	S2			
	S1	MST 2003		MST 2003
Thursday	S6			
	S5			
	S4			
	S3	DCS 3003	MST 1003	
	S2			MST 2003
	S1			
Friday	S6			
	S5			
	S4			
	S3			
	S2			DCS 3003
	S1			

HTML – ActivityLesson 9 (Challenging)

https://qian04.github.io/qianyuqian.github.io/activitylesson9(challenging)

https://kooxuelun.github.io/ActivityLesson9(challenging).html

https://tanjiaming2.github.io/activitylesson9.html

https://andyhoweiyan.github.io/Crecendo%20Time%20table.html

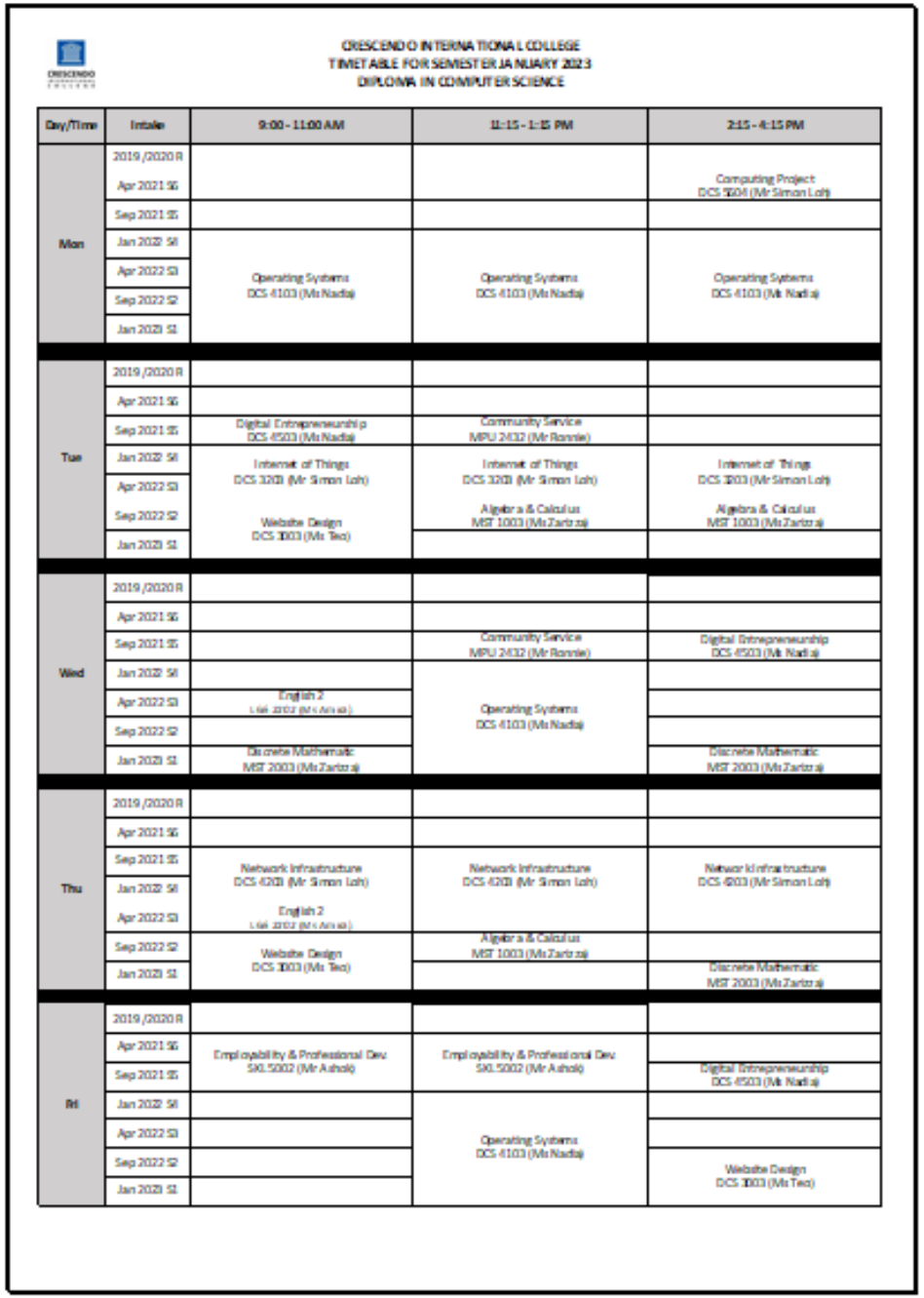
https://tanjunxu.github.io/activitylesson9(challenging).html

https://choobenghao.github.io/ActivityLesson9_Challenging.html

HTML –

ActivityLesson 9

(Tough)



CRESCENDO INTERNATIONAL COLLEGE
TIMETABLE FOR SEMESTER JANUARY 2023
DIPLOMA IN COMPUTER SCIENCE

Day/Time	Intake	9:00 - 11:00 AM	11:15 - 1:15 PM	2:15 - 4:15 PM
Mon	2019/2020 R			
	Apr 2021 S6			Computing Project DCS 5604 (Mr Simon Loh)
	Sep 2021 S5			
	Jan 2022 S4	Operating Systems DCS 4103 (Ms Nadia)	Operating Systems DCS 4103 (Ms Nadia)	Operating Systems DCS 4103 (Ms Nadia)
	Apr 2022 S3			
	Sep 2022 S2			
	Jan 2023 S1			
Tue	2019/2020 R			
	Apr 2021 S6			
	Sep 2021 S5	Digital Entrepreneurship DCS 4503 (Ms Nadia)	Community Service MPU 2432 (Mr Ronnie)	
	Jan 2022 S4	Internet of Things DCS 3203 (Mr Simon Loh)	Internet of Things DCS 3203 (Mr Simon Loh)	Internet of Things DCS 3203 (Mr Simon Loh)
	Apr 2022 S3		Algebra & Calculus MST 1003 (Ms Zarizaj	Algebra & Calculus MST 1003 (Ms Zarizaj
	Sep 2022 S2	Website Design DCS 3303 (Ms Teo)		
	Jan 2023 S1			
Wed	2019/2020 R			
	Apr 2021 S6			
	Sep 2021 S5		Community Service MPU 2432 (Mr Ronnie)	Digital Entrepreneurship DCS 4503 (Ms Nadia)
	Jan 2022 S4			
	Apr 2022 S3	English 2 Unit 3202 (Ms Anisa)	Operating Systems DCS 4103 (Ms Nadia)	
	Sep 2022 S2			
	Jan 2023 S1	Discrete Mathematic MST 2003 (Ms Zarizaj		Discrete Mathematic MST 2003 (Ms Zarizaj
Thu	2019/2020 R			
	Apr 2021 S6			
	Sep 2021 S5	Network Infrastructure DCS 4203 (Mr Simon Loh)	Network Infrastructure DCS 4203 (Mr Simon Loh)	Network Infrastructure DCS 4203 (Mr Simon Loh)
	Jan 2022 S4			
	Apr 2022 S3	English 2 Unit 3202 (Ms Anisa)		
	Sep 2022 S2	Website Design DCS 3303 (Ms Teo)	Algebra & Calculus MST 1003 (Ms Zarizaj	
	Jan 2023 S1			Discrete Mathematic MST 2003 (Ms Zarizaj
Fri	2019/2020 R			
	Apr 2021 S6	Employability & Professional Dev SKL 5002 (Mr Ashok)	Employability & Professional Dev SKL 5002 (Mr Ashok)	
	Sep 2021 S5			Digital Entrepreneurship DCS 4503 (Ms Nadia)
	Jan 2022 S4		Operating Systems DCS 4103 (Ms Nadia)	
	Apr 2022 S3			
	Sep 2022 S2			Website Design DCS 3303 (Ms Teo)
	Jan 2023 S1			

HTML – ActivityLesson 9 (Tough)

https://tanjunxu.github.io/activitylesson9(tough).html

https://rrachel24.github.io/activitylesson9.html

HTML

Table + JavaScript

HTML –Table + JavaScript

*activity13i_3 - Notepad

File Edit Format View Help

```
<html><head><title>Table</title></head>
<body bgcolor="azure"
leftmargin="100" topmargin="90" >
<table border=1>
<tr>
<th>header one</th>
<th>header two</th>
<th>header three</th>
<th>header four</th>
</tr>

<script>
function myFunction() {
  alert("data row1-col1");
}
function myFunction2() {
  alert("data row1-col2");
}
function myFunction3() {
  alert("data row1-col3");
}
</script>
```

*activity13i_3 - Notepad

File Edit Format View Help

```
<tr>
<td id="demo" onclick="myFunction()">
data row1-col1</td>
<td id="demo2" onmouseover="myFunction2()">
data row1-col2</td>
<td id="demo3" ondblclick="myFunction3()">
data row1-col3</td>
<td>data row1-col4</td>
</tr>

<tr>
<td>data row2-col1</td>
<td>data row2-col2</td>
<td>data row2-col3</td>
<td>data row2-col4</td>
</tr>

</table>
</body></html>
```

HTML

Form

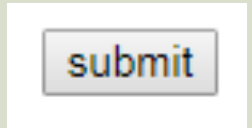
HTML – Form

1) An HTML form is used to collect data from the site visitor.
2) Web forms are one of the main points of interaction between a user and a website or application.
3) The user input is most often sent to a server for processing, input will post to a back-end application such as CGI, ASP Script or PHP script etc.
4) The back-end application will perform required processing on the passed data.

HTML – Form Types

Type	Description	Tag	Example
text	to input single-line text	<input type=text name=username>	Username:
text area	to input multiple-line text	<textarea rows=3 cols=30 name=address></textarea>	Address:
password	similar to text field, characters are shown as asterisks or dots	<input type=password name=user-password>	Password : •••••••••
checkbox	to select multiple options	<input type=checkbox name=>	Select at Least 2 Interests JavaScript HTML CSS Progressive Web Apps

HTML – Form Types

Type	Description	Tag	Example
radio	provides a set of options	<input type = radio name = Gender value =Male>	
button	to define a clickable button	<input type=button>	
submit	to submit all form values to a form-handler	<input type=submit>	
reset	to reset the form to the default values	<input type=reset>	
date	to enter a date values	<input type=date>	

HTML – Activity 15g

*activity15g - Notepad

File Edit Format View Help

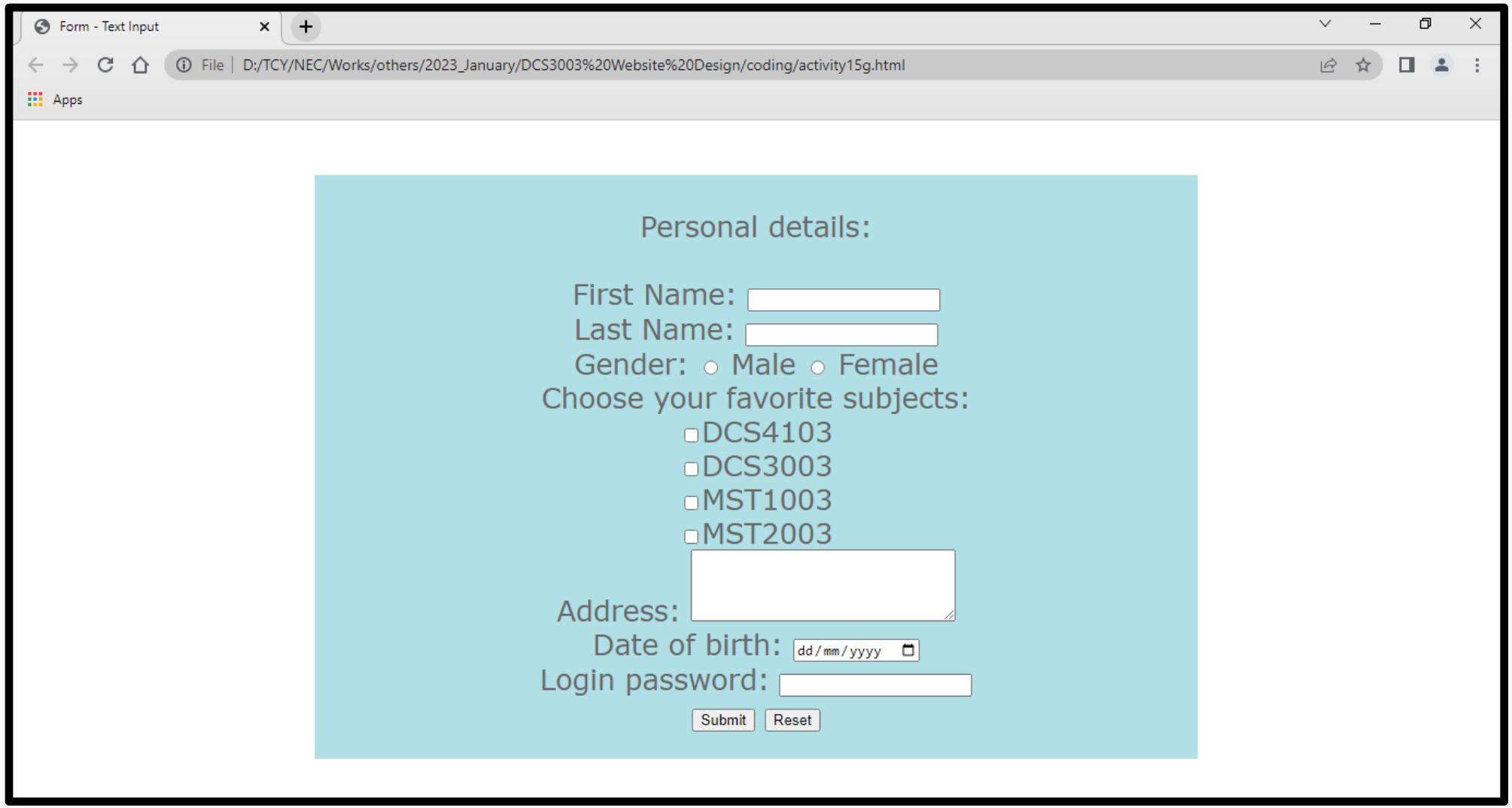
```
<html><head><title>Form - Text Input</title></head>
<body topmargin=50>
<center><form>

<p style="background-color:powderblue;color:#696969;
font-family:verdana;font-size:160%;
text-align:center;width:fit-content;">

<br>Personal details:<br><br>
First Name: <input type=text name=fname><br>
Last Name: <input type=text name=lname><br>
Gender:
<input type = radio name = gender value = Male> Male
<input type = radio name = gender value = Female> Female
<br>
Choose your favorite subjects:<br>
<input type=checkbox name=fav_sub value=DCS4103>
DCS4103<br>
```

*activity15g - Notepad

File Edit Format View Help

```
<input type=checkbox name=fav_sub value=DCS3003>
DCS3003<br>
<input type=checkbox name=fav_sub value=MST1003>
MST1003<br>
<input type=checkbox name=fav_sub value=MST2003>
MST2003<br>
Address:
<textarea rows=4 cols=30 name=address></textarea><br>
Date of birth:
<input type=date name=dob>
<br>
Login password:
<input type=password name=loginpass>
<br>
<input type=submit value=Submit>
<input type=reset value=Reset>
</form></center>
</body></html>
```

HTML – Activity 15g



HTML – Form

HOW CAN THE FORM BE MADE TO LOOK NEAT AND ORDERLY?

HTML – Form

HOW CAN THE FORM BE MADE TO LOOK NEAT AND ORDERLY?

- <Table>
- CSS

HTML

Form (cont.)

HTML – Form

HOW CAN THE FORM BE MADE TO LOOK NEAT AND ORDERLY?

- 1st step: use <Table> tag to design form's layout

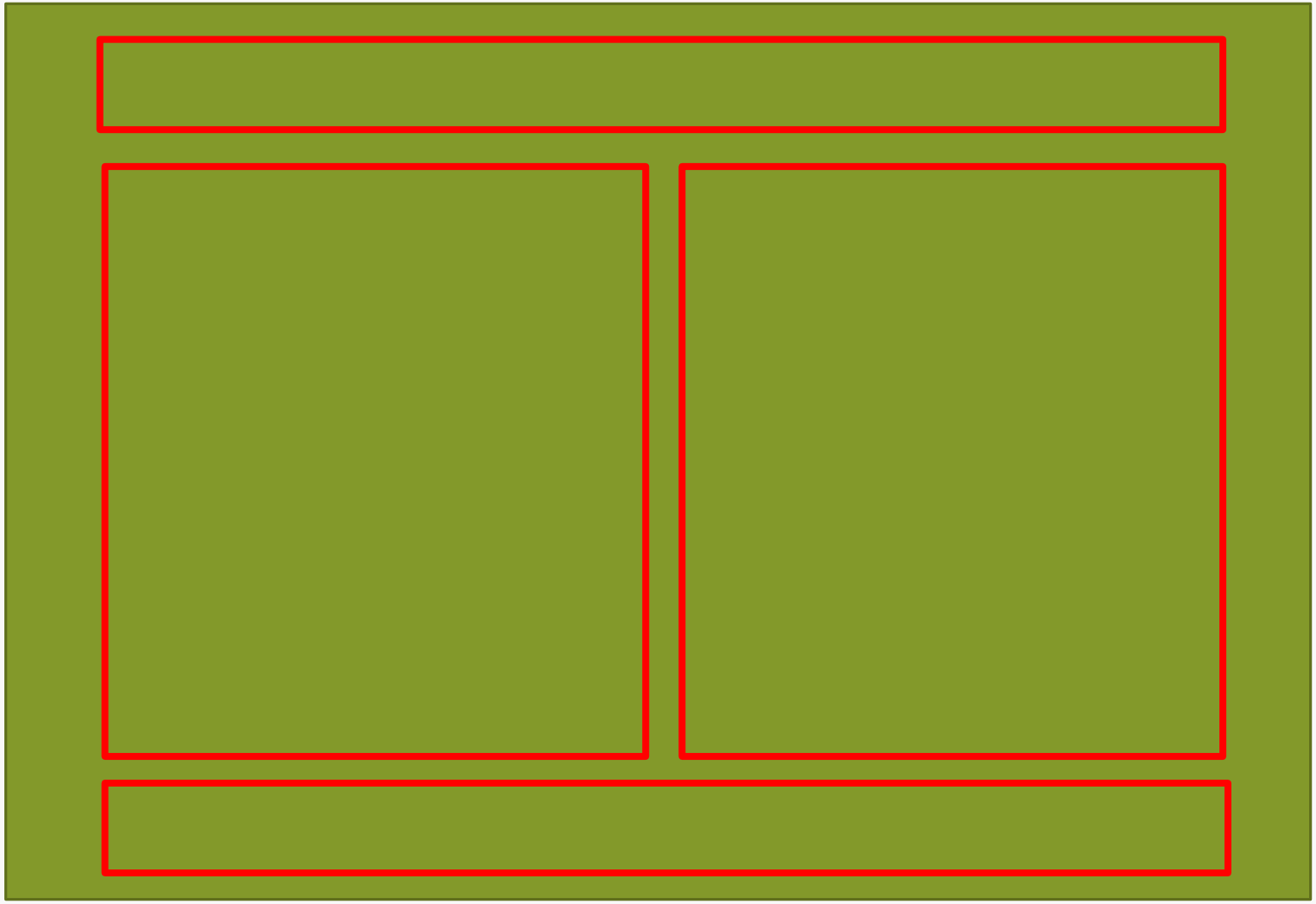
HTML – Table - Tags

Tag	Description
<table>	It defines a table.
<tr>	It defines a row in a table.
<th>	It defines a header cell in a table.
<td>	It defines a cell in a table.
<caption>	It defines the table caption.
<colspan>	It defines the number of columns a table cell should span.
<rowspan>	It specifies the number of rows a cell should span.

HTML – **Activity 15h**

Example:

1) Design a table layout
2) Insert the form's content into <td>



HTML

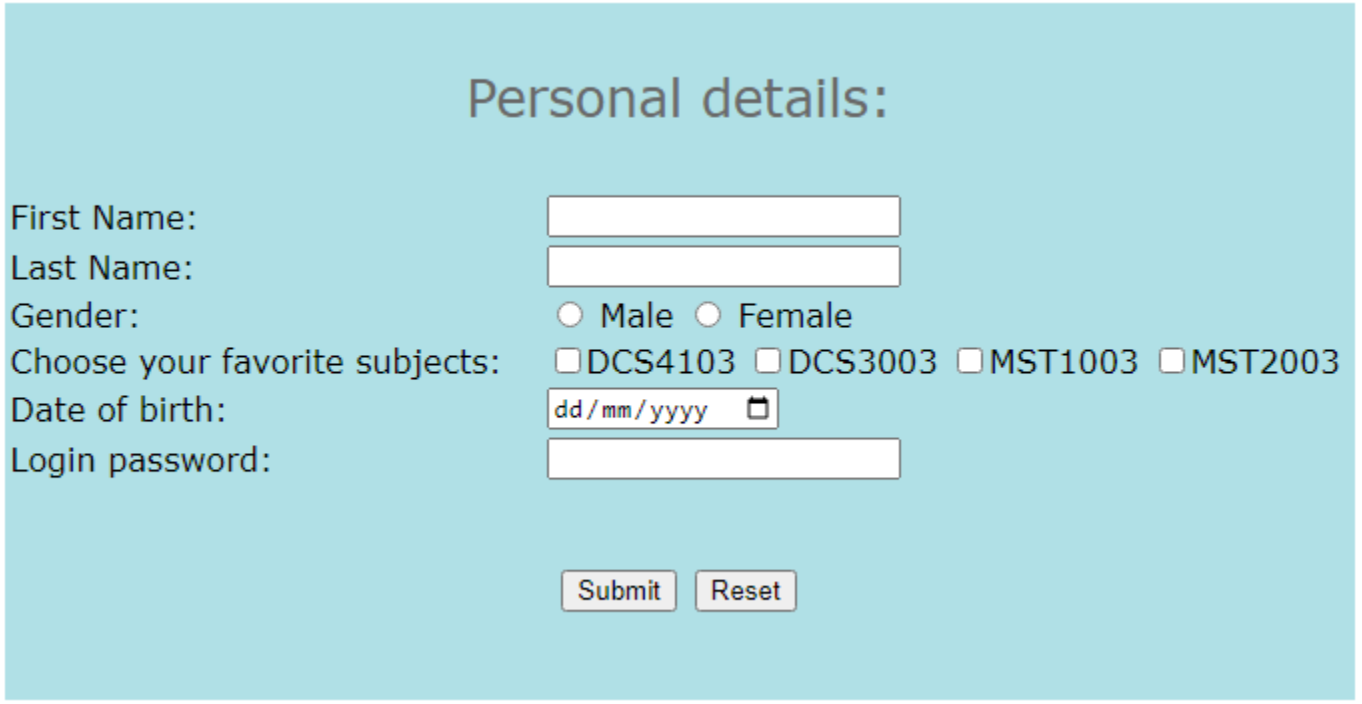
Quiz 2

Use <Table> tag to design the form's layout

HTML – Activity 15h

Before <table>

After <table>



HTML – Activity 15h - <table>

activity15h - Notepad

File Edit Format View Help

```
<html><head><title>Form - Text Input</title></head>
<body topmargin=50><center><form>
<p style="background-color:powderblue;color:#696969;
font-family:verdana;font-size:160%;text-align:center;
width:fit-content;">
<br>Personal details:<br><br>
<table><tr>
<td width='40%'>First Name: </td>
<td><input type=text name=fname></td>
</tr><tr>
<td>Last Name: </td>
<td><input type=text name=lname> </td>
</tr><tr>
<td>Gender: </td>
<td>
<input type = radio name = gender value = Male> Male
<input type = radio name = gender value = Female> Female
</td>
</tr><tr>
<td>Choose your favorite subjects:</td>
```

activity15h - Notepad

File Edit Format View Help

```
<td>
<input type=checkbox name=fav_sub value=DCS4103>DCS4103
<input type=checkbox name=fav_sub value=DCS3003>DCS3003
<input type=checkbox name=fav_sub value=MST1003>MST1003
<input type=checkbox name=fav_sub value=MST2003>MST2003
</td>
</tr><tr>
<td>Date of birth:</td>
<td><input type=date name=dob></td>
</tr><tr>
<td>Login password:</td>
<td><input type=password name=loginpass></td>
</tr>
</table>
<br>
<input type=submit value=Submit>
<input type=reset value=Reset>
</td></tr>
<br><br>
</form></center></body></html>
```

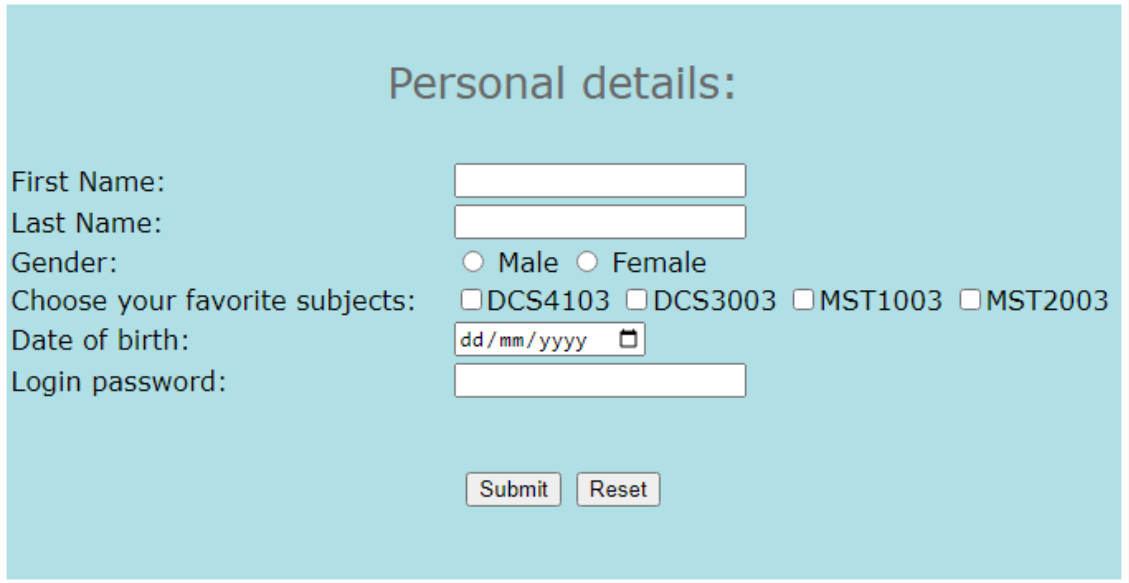
HTML – Form

HOW CAN THE FORM BE MADE TO LOOK NEAT AND ORDERLY?

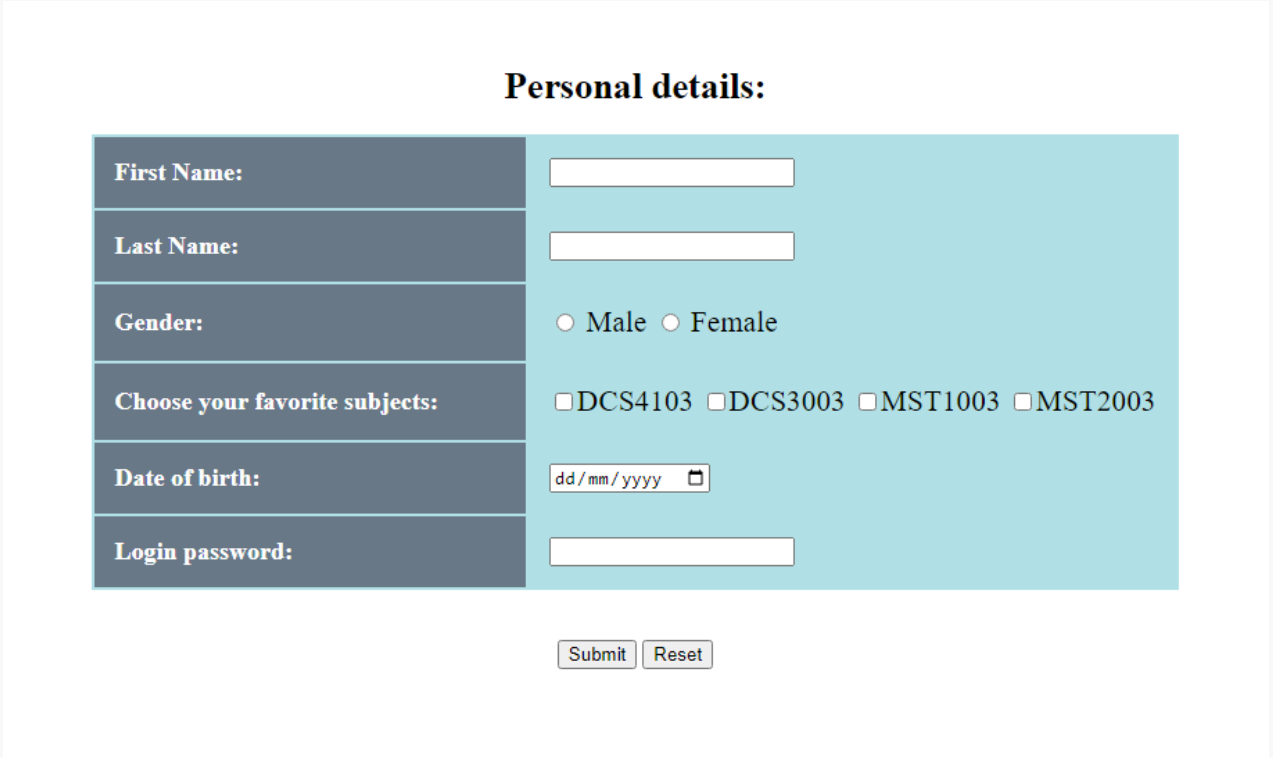
- 2nd step: use CSS to design the look and formatting of the form.

HTML – Activity 15i

Before CSS



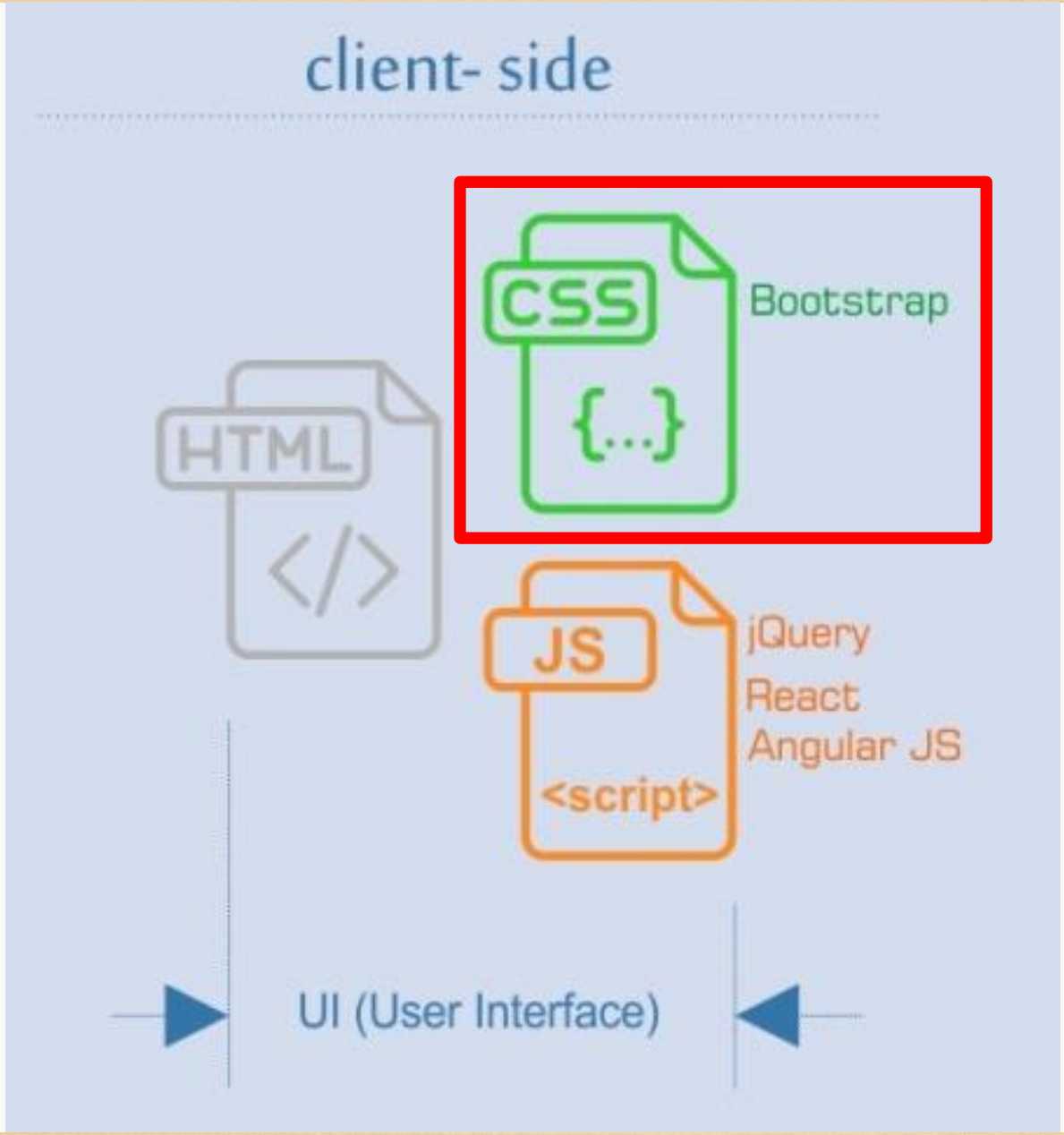
After CSS



CSS

Introduction

DCS 3003 Website Design



DCS 3003 Website Design

CSS

1) Stands for **C**ascading **S**tyle **S**heets.
2) In short, CSS is a design language that makes a website look more appealing than just plain or uninspiring pieces of text.
3) CSS determines visual structure, layout, and aesthetics.

CSS

1) Cascading Style Sheet (CSS) is used to set the style in web pages that contain HTML elements.

2) With CSS, we can assign

- Font properties
- Sizes
- Background images
- Color
- Borders
- Position of elements

CSS – The advantages:

CSS saves time

- Write CSS once and then reuse same sheet in multiple HTML pages.

Pages load faster

- Just write one CSS rule of a tag and apply to all the occurrences of that tag. So less code means faster download times.

Easy maintenance

- To make a global change, change the style and all elements in all the web pages will be updated automatically, to standardize the coding style.

Superior styles to HTML

- CSS has a much wider array of attributes than HTML so you can give far better look to your HTML page in comparison of HTML attributes.

Multiple device compatibility

- style sheets allow content to be optimized for more than one type of device such as PDAs and cell phones.

Global web standards

- now HTML attributes are being deprecated and it is being recommended to use CSS.

CSS

In HTML, different objects have different styles of coding or different property for the same presentation settings.

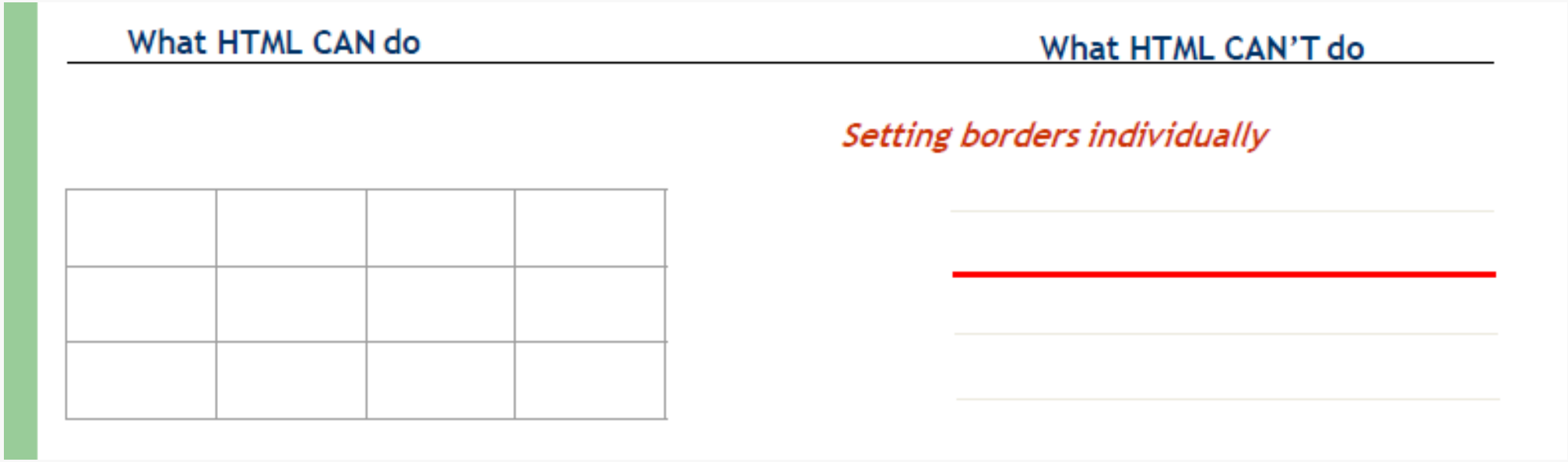
-> Example : setting the color of table borders and text are different

```
<table border="1" bordercolor="red">
……..
</table>


<font color="red">
 ……..
</font>
```

CSS

CSS can extend the attributes and features of HTML Tags because <u>HTML tag has limited attributes</u>

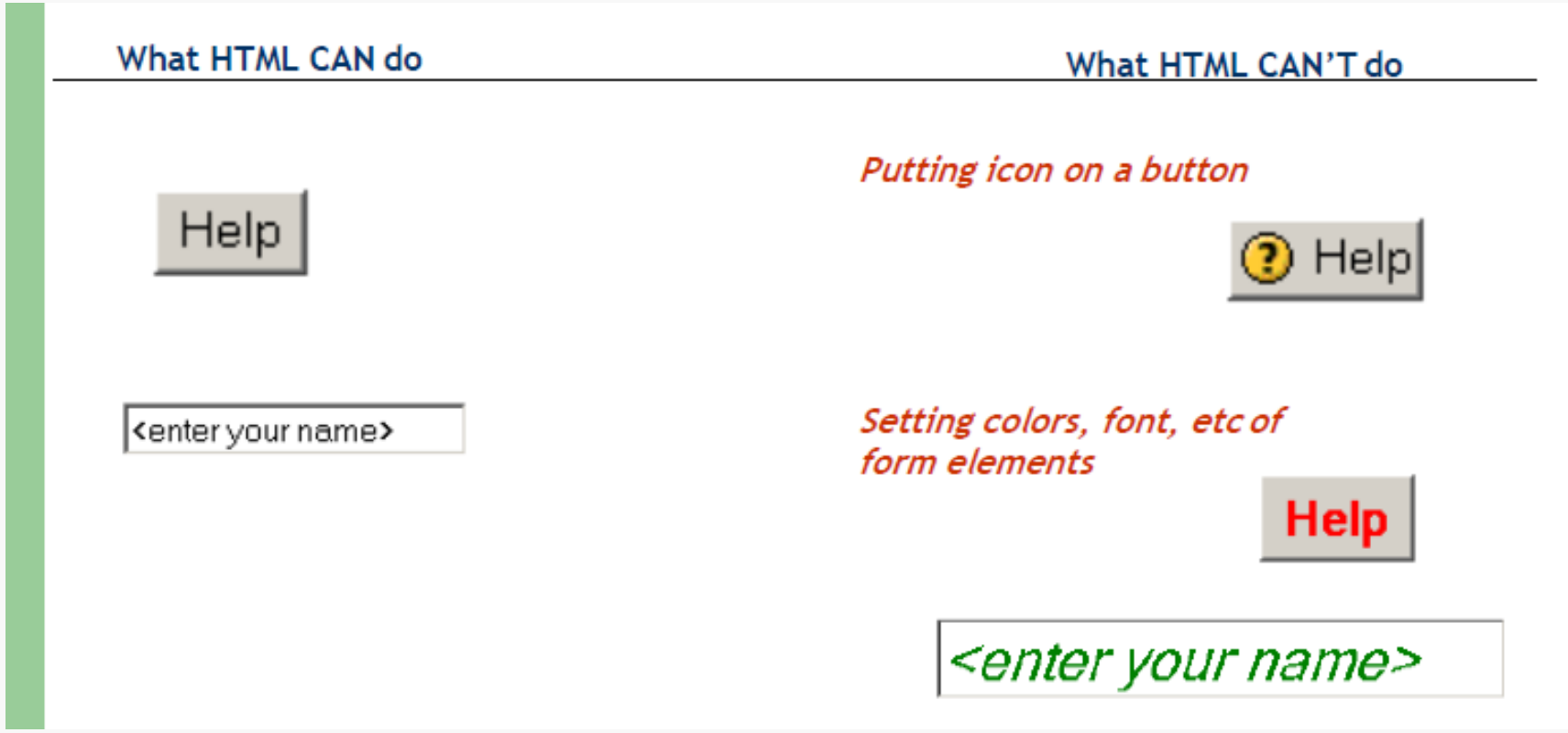
-> Example :



CSS

CSS can extend the attributes and features of HTML Tags

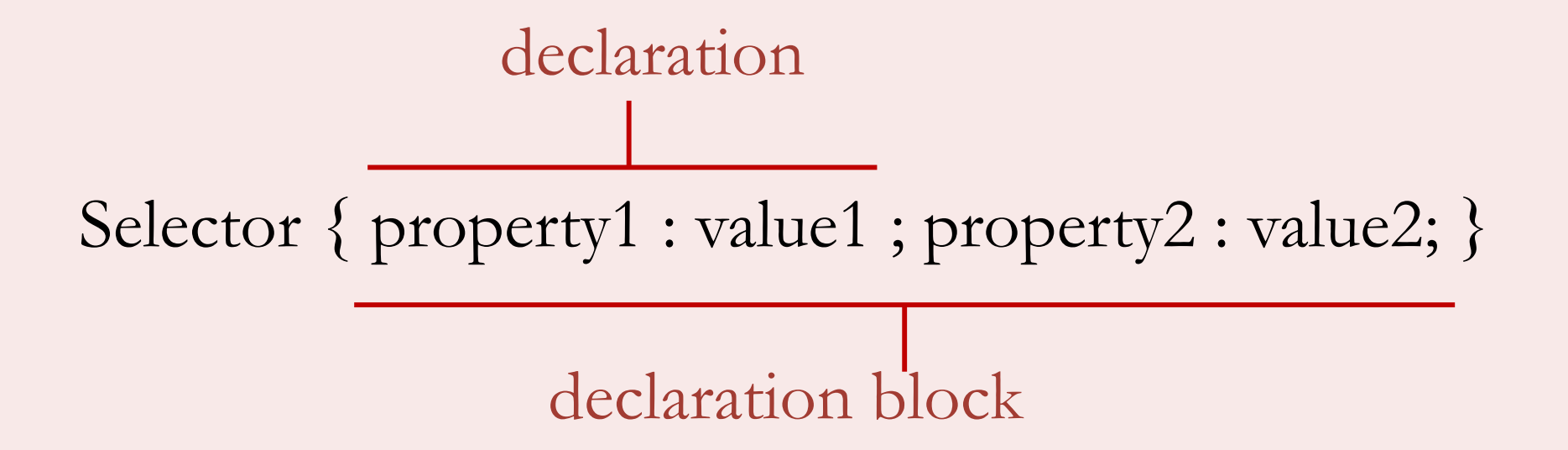
-> Example :



Syntax

-> a rule consists of a selector, followed by property and value.

-> each pair called declaration or style

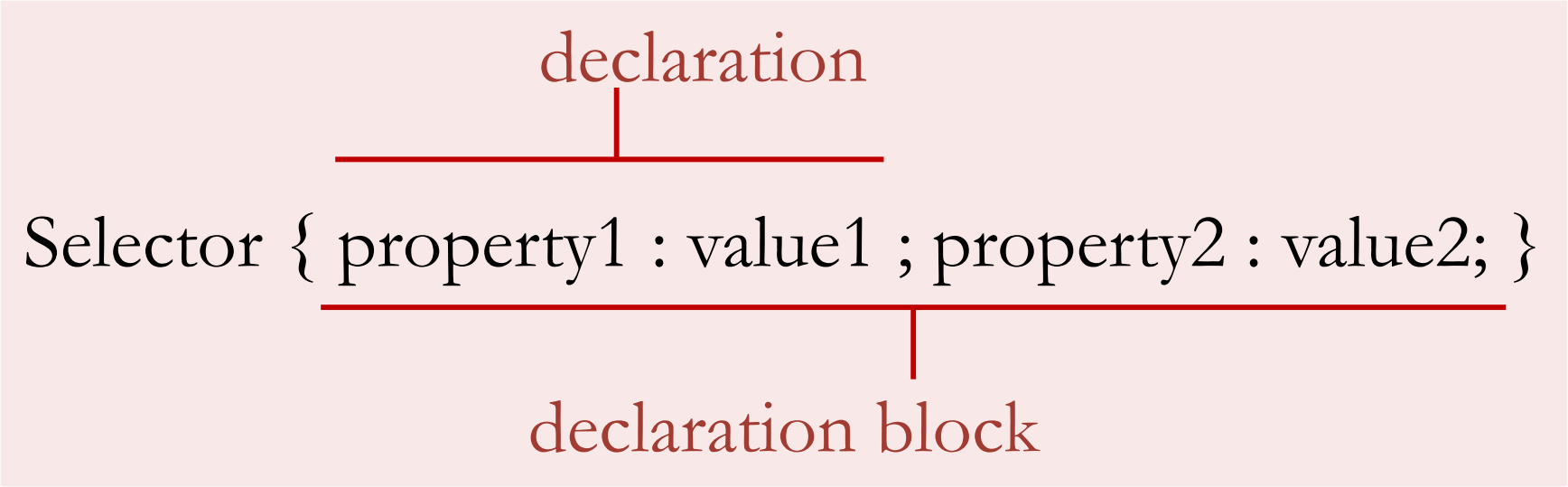


Syntax (Ruleset)

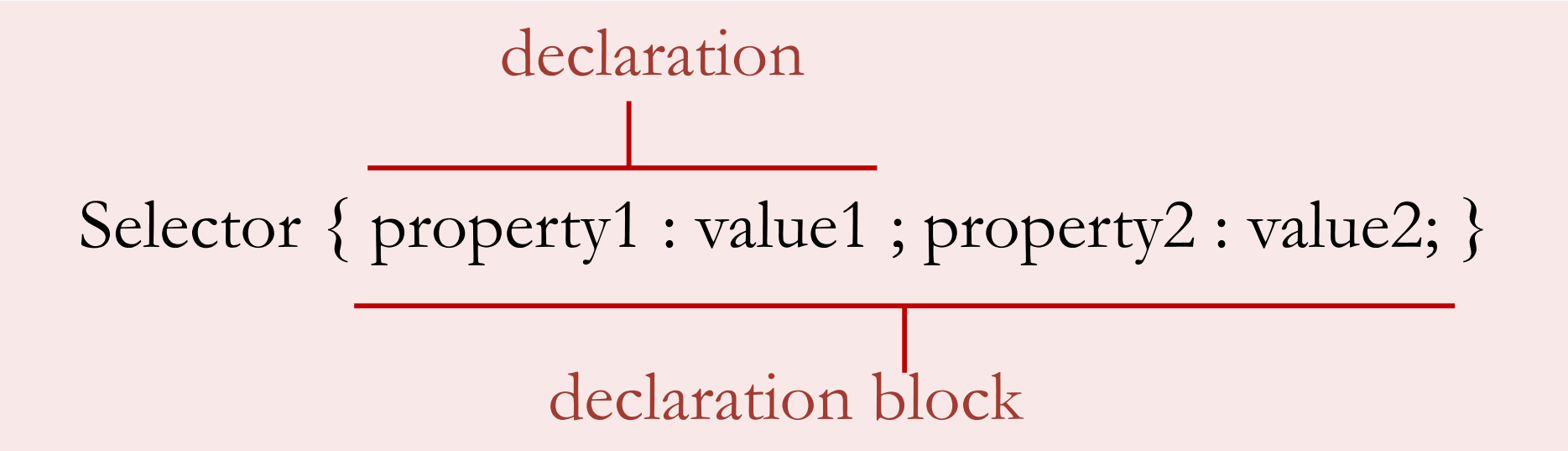
-> Each ruleset must be wrapped in curly braces ({}).

-> each declaration, colon (:) is use to separate the property and value.

-> multiple sets of CSS declarations separate by a semicolon (;)

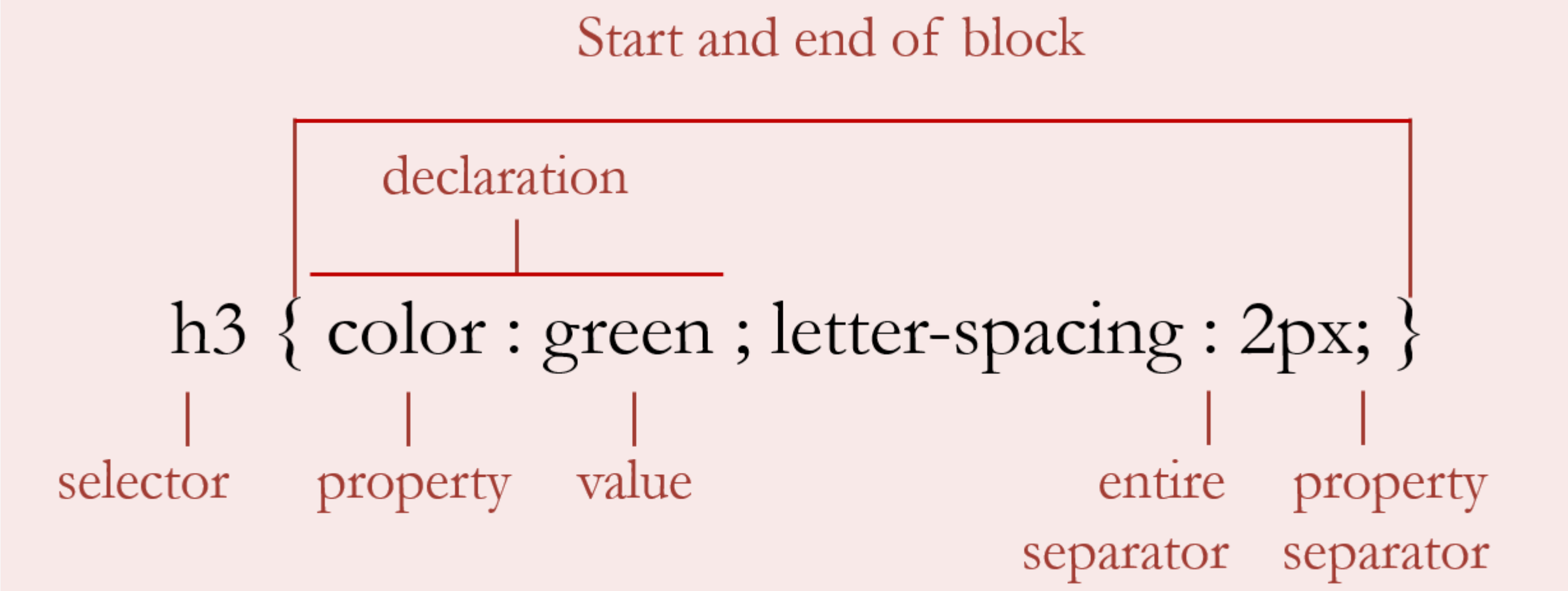


Syntax - CSS rule set has 3 parts:

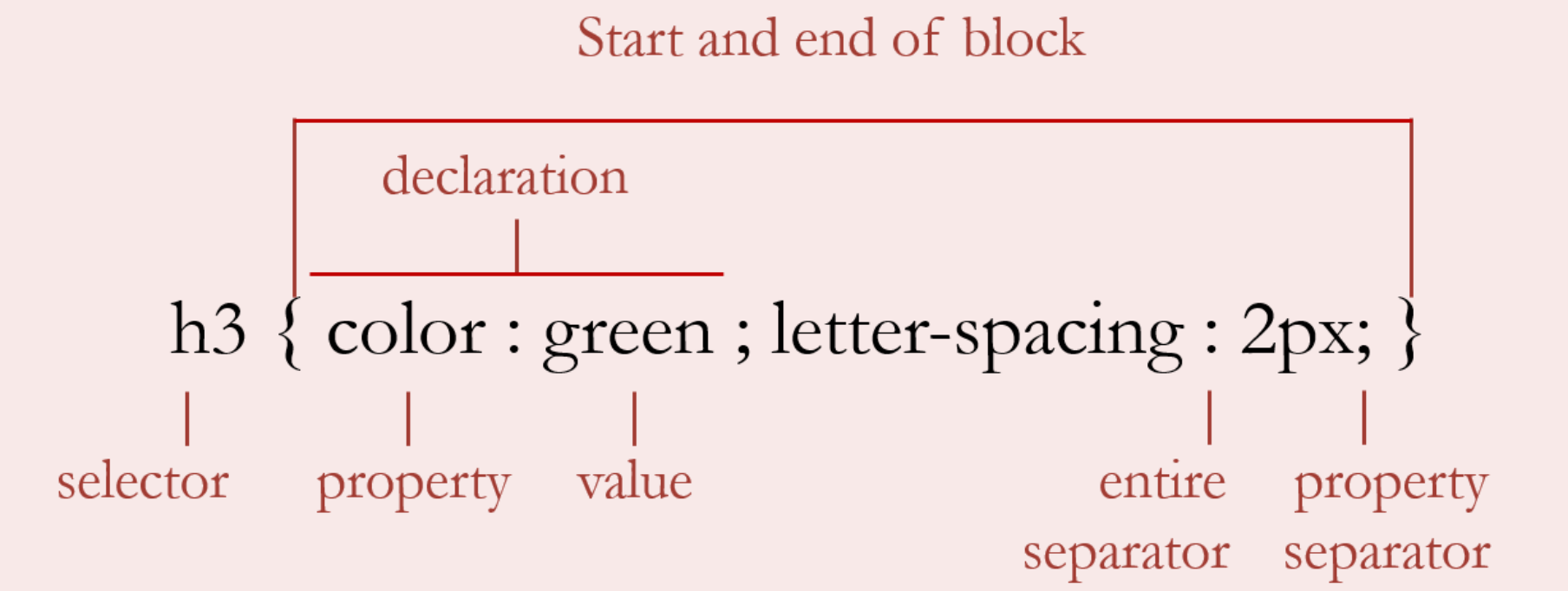


1. **Selector**
2. **Property**
3. **Value**

Syntax - CSS rule set has 3 parts: **Example**

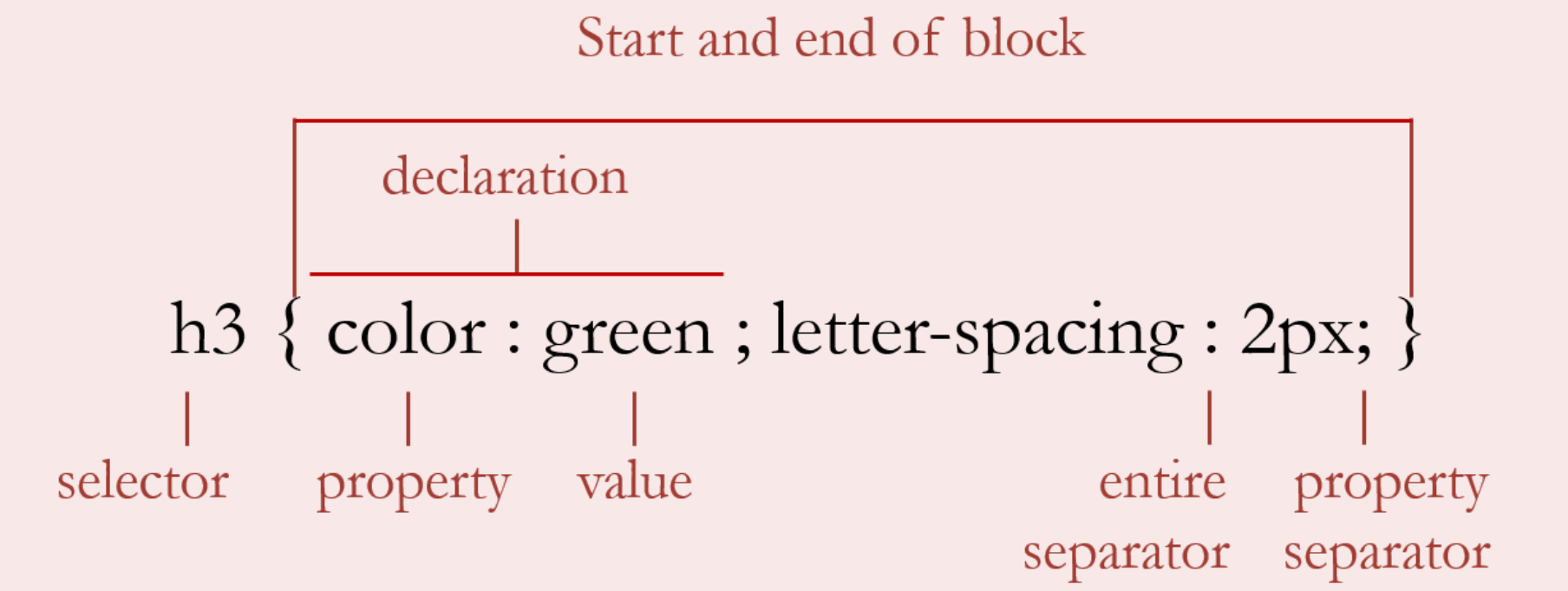


Syntax - CSS rule set has 3 parts:



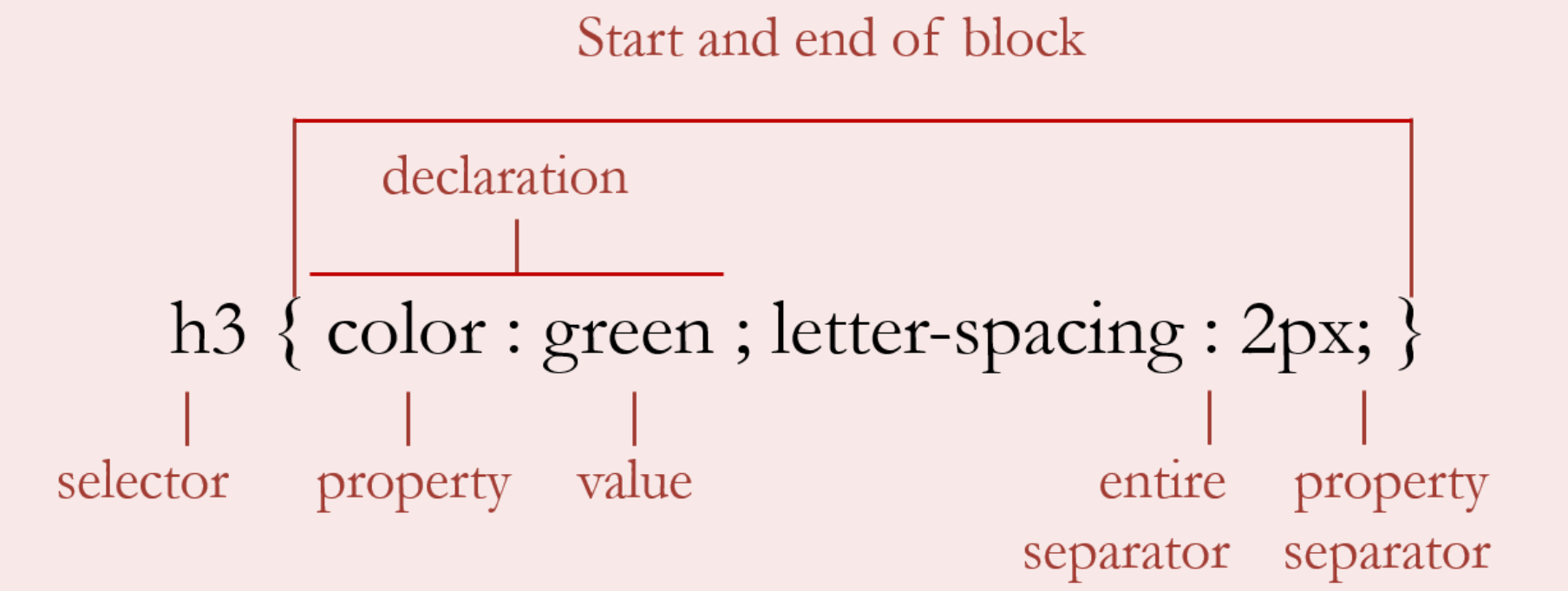
Selector - A selector is an HTML tag at which a style will be applied. This could be any tag like <h1> or <p> etc.

Syntax - CSS rule set has 3 parts:



Property - A property is a type of attribute of HTML tag. Put simply, all the HTML attributes are converted into CSS properties. They could be color, border etc.

Syntax - CSS rule set has 3 parts:



Value - Values are assigned to properties. For example, color property can have value either green or #FFFFFF etc.

CSS – Selector Types

1) Tag/Element
2) Identifaction (id)
3) Class
4) Tag-spesific
5) Grouping
6) Contextual
7) Pseudo-class

CSS – Selector Types - Tag/Element

- Using name of element/tag
- To redefine the default value of tags

Example:

Define

```
p {color:red;}   /*makes all p tags red*/
h1 {font-size: 20px;}/*makes all text set by h1 tag to be 20 pixel in size*/
```

Apply

```
<h1> <p> This is a red text with 20 pixel in size</p></h1>
```

CSS – Selector Types - Identifaction (id)

- To define styles, the selector names use prefix # character
- To apply the styles, set the ID attribute with the name

Example:

Define

```
#red {color:red;}
```

Apply

```
<b id="red"> This is a red bold text </b>
```

CSS – Selector Types - Class

- To define styles, the selector names use prefix . (dot) character
- To apply the styles, set the CLASS attribute with the name

Example:

Define

```
.green {color:#00FF00; font-weight:bold;}
```

Apply

```
<font class="green"> I like green color</font>
```

CSS – Selector Types - Tag-Specific

- Style can only be applied to elements which defined it.
- Selector name: tag_name.style_name

Example:

Define

```
font.redbold {color:red; font-weight:bold;}
```

Apply

```
<font class="redbold"> This is a red bold text</font>
<b class="redbold"> redbold style has no effect here</b>
```

CSS – Selector Types - Grouping

- Define the same styles to a group of tags.

Example:

Define

```
h1,h2,h3 {background-color: blue;}
/*sets the background color of all h1,h2,and h3 elements to blue */
```

Apply

```
<h1> This is heading type1 with blue background color</h1>
<h2> This is heading type2 with blue background color</h2>
<h3> This is heading type3 with blue background color</h3>
```

CSS – Example:

Grouping selectors

```
h1,h2,h3 {
  color: cyan;
  padding: 15px;
  background: silver;
}
```

CSS – Example:

Grouping selectors

```
p,li,h1 {
  color: cyan;
  padding: 15px;
  background: silver;
}
```

CSS – Selector Types - Contextual

- Applies to descendent tags.

Example:

Define

```
p b{color:red; text-decoration: underline;}
```

Apply

```
<b>CSS has no effect here</b>
<p>CSS has no effect here</p>
<p>
    <b>CSS has effect here. This is a red underlined text</b>
</p>
```

CSS – Selector Types - Pseudo-class

- CSS pseudo-classes are used to add styles to selectors, but only when those selectors meet certain conditions.
- A pseudo class is expressed by adding a **colon (:)** after a selector.

Example:

a:link -> specifies the unvisited links

a:hover -> specifies the link as mouse cursor is hovering on it

a:active -> specifies the link as it is being clicked

a:visited -> specifies the link after being clicked

CSS – Selector Types - Pseudo-class

- CSS pseudo-classes are used to add styles to selectors, but only when those selectors meet certain conditions.
- A pseudo class is expressed by adding a **colon** (:) after a selector

Example:

Define

```
a:link {font-weight: bold;} /* makes unvisited links bold */
a:hover {text-transform: uppercase;} /* makes links uppercase as mouse
cursor is hovering on*/
a:active {color: red;} /* makes links red as they are clicked */
a:visited {text-decoration: underline;} /* makes visited links underlined*/
```

CSS – Example:

Type 1 →

```
selector {
    property1: value1;
    property2: value2;
    property3: value3
/* last semicolon is optional */
}
```

Type 2 →

```
selector {property1: value1; property2: value2; property3: value3}
```

CSS – Example:

Type 1 →

```
h1 {
  color: cyan;
  padding: 15px;
  background: silver;
}
```

Type 2 →

```
h1 { color: cyan; padding: 15px; background: silver; }
```