

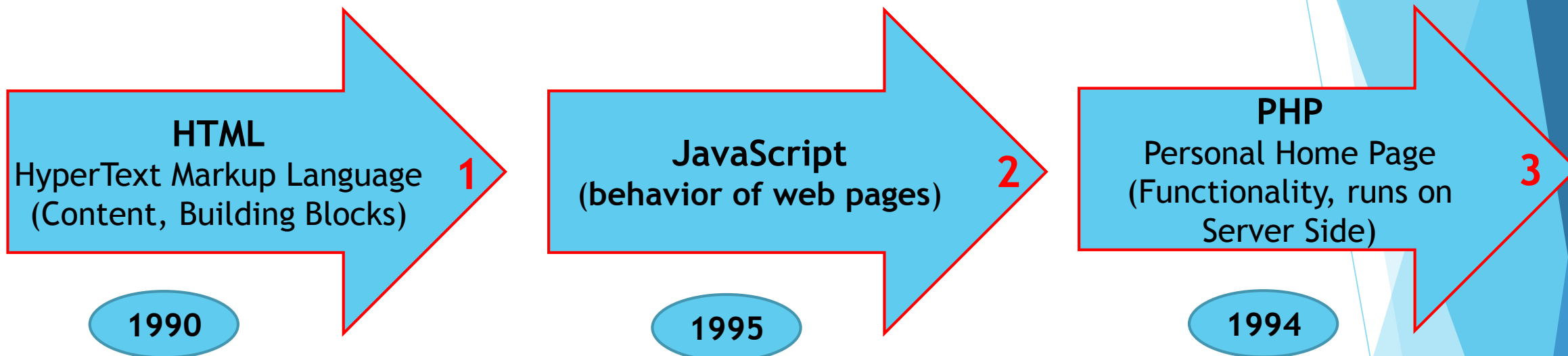


# الجامعة التقنية الجنوبية المعهد التقني القرنة قسم تقنيات أنظمة الحاسوب

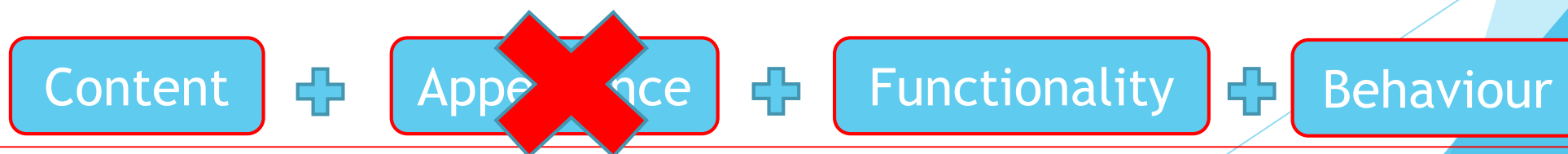
اساسيات تصميم المواقع CST106

محاضرة رقم (١) - HTML

# Road Map



- ▶ HTML is the language for building web pages.
- ▶ ~~CSS~~ is a language that describes the style of an HTML document.
- ▶ PHP is a server scripting language, and a powerful tool for making dynamic and interactive Web pages.
- ▶ JavaScript is a client scripting language, and it is used to program the behavior of web pages.



# HTML Introduction

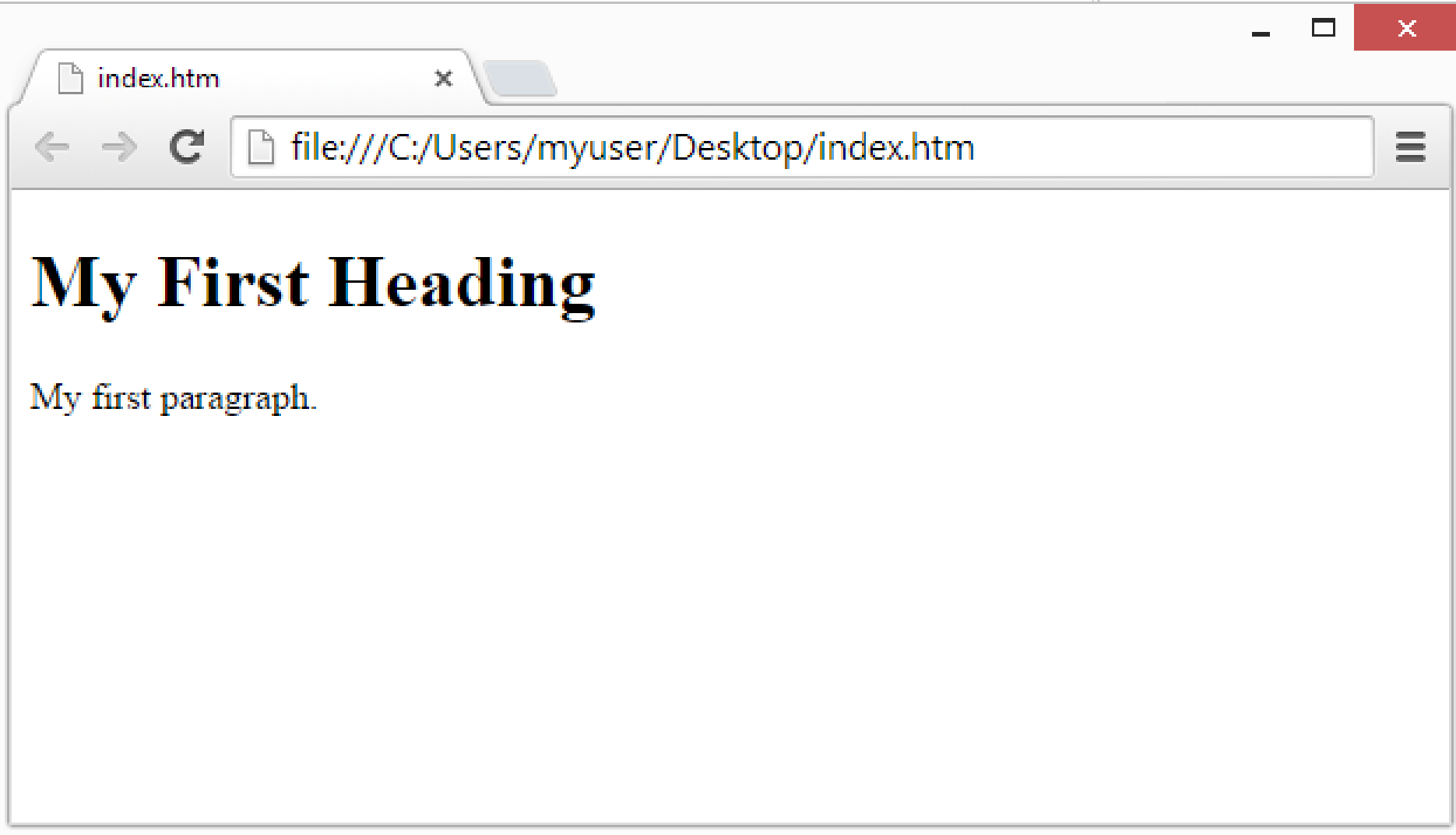
- ▶ HTML sta
- ▶ HTML des
- ▶ HTML con

defines t

contains  
about

contain

defin



an HTML

for the

heading

# HTML Elements (Tags)

- ▶ An HTML element usually consists of a **start tag** and an **end tag**, with the **content** inserted in between.

```
<tag-name>          Content goes here...          </tag-name>
```

Start tag	Element content	End tag
<h1>	My First Heading	</h1>
<p>	My first paragraph.	</p>

- ▶ An HTML element with no content is called **empty element** where it does not have an end tag.
- ▶ All HTML elements can have **attributes** which provide additional information about an element.

```
<p style="color:red"> This is a red paragraph. </p>
```

- ▶ The style attribute is used to specify the styling of an element, like color, font, size etc.
- ▶ Attributes are always specified in the start tag.

# HTML Elements (Tags)

- ▶ An HTML element usually consists of a **start tag** and an **end tag**, with the **content** inserted in between.

```
<tag-name>          Content goes here...          </tag-name>
```

Start tag	Element content	End tag
<h1>	My First Heading	</h1>
<p>	My first paragraph.	</p>

- ▶ An HTML element with no content is called **empty element** where it does not have an end tag.
- ▶ All HTML elements can have **attributes** which provide additional information about an element.

```
<p style="color:red"> This is a red paragraph. </p>
```

- ▶ The style attribute is used to specify the styling of an element, like color, font, size etc.
- ▶ Attributes are always specified in the start tag.

# HTML Headings

- ▶ Headings are defined with the `<h1>` to `<h6>` tags.
- ▶ `<h1>` defines the most important heading.
- ▶ `<h6>` defines the least important heading.
- ▶ `<hr>` is most often displayed as a horizontal rule.
- ▶ The HTML `<br>` element defines a line break.

# Heading 1

## Heading 2

### Heading 3

#### Heading 4

##### Heading 5

###### Heading 6

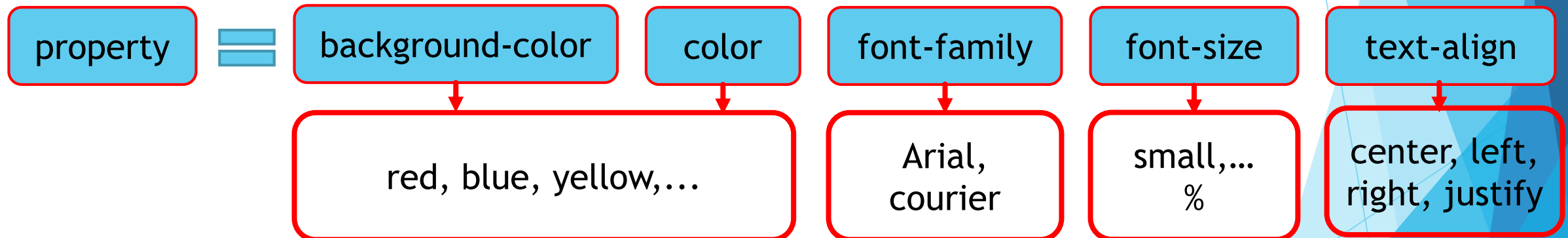
```
<!DOCTYPE html>
<html>
<head>
<title>
Page Title
</title>
</head>
<body>
<h1>Heading 1</h1>
<h2>Heading 2</h2>
<h3>Heading 3</h3>
<h4>Heading 4</h4>
<h5>Heading 5</h5>
<h6>Heading 6</h6>
</body>
</html>
```

# HTML Styles

- ▶ Setting the style of an HTML element, can be done with the style attribute.
- ▶ The HTML style attribute has the following syntax:

```
<tag-name style = "property : value;">
```

- ▶ The **property** is a CSS property and the **value** is a CSS value.



```
<p style = "text-align : center;"> Qurna TI </p>
```

```
<p style = "font-family : courier;"> Qurna TI </p>
```



# الجامعة التقنية الجنوبية المعهد التقني القرنة قسم تقنيات أنظمة الحاسوب



تصميم المواقع الالكترونية

محاضرة رقم (٢) - PHP



# PHP Numbers

PHP numbers	Definition	Example	Range	Functions used to check type
<b>Integer</b>	a number without any decimal part	-256, 10358	-2147483648 to 2147483647	is_int(), is_integer(), is_long()
<b>Float</b>	a number with a decimal point or in exponential form	2.0, 6.4, 7.6E+5	Up to 1.7976931348623 E+308	is_float() & is_double()
<b>Infinity</b>	a numeric value larger than PHP_FLOAT_MAX	1.9E411	Greater than 1.7976931348623 E+308	is_finite() & is_infinite()
<b>NaN</b>	used for impossible mathematical operations	log(-1)	Not a Number	is_nan()

- ▶ PHP **is\_numeric()** function can be used to find whether a variable is numeric.
- ▶ **is\_numeric()** function returns true if the variable is a number or a numeric string, false otherwise.

# PHP Math

Math function	Usage	Example	Return
<b>pi()</b>	returns the value of PI	pi()	3,141592653589 Λ
<b>min()</b>	finds the lowest value in a list of arguments	min(0, 150, 20, -8, -200)	-200
<b>max()</b>	finds the highest value in a list of arguments	max(0, 150, 20, -8, -200)	150
<b>abs()</b>	returns the absolute value of a number	abs(-1)	1
<b>sqrt()</b>	returns the square root of a number	sqrt(9)	3
<b>round()</b>	rounds a number to its nearest integer	round(9.3)	9
<b>rand()</b>	generates a random number	rand(), rand(10, 100)	1281167960, 97

# PHP Arithmetic Operators

Operator	Name	Example	Result
+	Addition	$\$x + \$y$	Sum of $\$x$ and $\$y$
-	Subtraction	$\$x - \$y$	Difference of $\$x$ and $\$y$
*	Multiplication	$\$x * \$y$	Product of $\$x$ and $\$y$
/	Division	$\$x / \$y$	Quotient of $\$x$ and $\$y$
%	Modulus	$\$x \% \$y$	Remainder of $\$x$ divided by $\$y$
**	Exponentiation	$\$x ** \$y$	Result of raising $\$x$ to the $\$y$ 'th power
++ $\$x$	Pre-increment	++ $\$x$	Increments $\$x$ by one, then returns $\$x$
$\$x$ ++	Post-increment	$\$x$ ++	Returns $\$x$ , then increments $\$x$ by one
-- $\$x$	Pre-decrement	-- $\$x$	Decrements $\$x$ by one, then returns $\$x$
$\$x$ --	Post-decrement	$\$x$ --	Returns $\$x$ , then decrements $\$x$ by one

- ▶ The return value of `is_numeric("353")` is:
  - True (1)
  - False (0)
- ▶ The return value of `is_infinite("32e400")` is:
  - True (1)
  - False (0)
- ▶ What is the output of the following code:

```
<?php
$a = 32;
$a++;
++$a;
--$a;
echo $a--;
?>
```

- 32
- 33
- 34

# Questions

# PHP Comparison Operators

Operator	Name	Example	Return
==	Equal	$\$x == \$y$	True if $\$x$ is equal to $\$y$
===	Identical	$\$x === \$y$	True if $\$x$ is equal to $\$y$ , and they are of the same type
!=	Not equal	$\$x != \$y$	True if $\$x$ is not equal to $\$y$
<>	Not equal	$\$x <> \$y$	True if $\$x$ is not equal to $\$y$
!==	Not identical	$\$x !== \$y$	True if $\$x$ is not equal to $\$y$ , or they are not of the same type
>	Greater than	$\$x > \$y$	True if $\$x$ is greater than $\$y$
<	Less than	$\$x < \$y$	True if $\$x$ is less than $\$y$
>=	Greater than or equal to	$\$x >= \$y$	True if $\$x$ is greater than or equal to $\$y$
<=	Less than or equal to	$\$x <= \$y$	True if $\$x$ is less than or equal to $\$y$

# PHP Array Operators

Operator	Name	Example	Result
+	Union	$\$x + \$y$	Union of $\$x$ and $\$y$
==	Equality	$\$x == \$y$	Returns true if $\$x$ and $\$y$ have the same key/value pairs
===	Identity	$\$x === \$y$	Returns true if $\$x$ and $\$y$ have the same key/value pairs in the same order and of the same types
!=	Inequality	$\$x != \$y$	Returns true if $\$x$ is not equal to $\$y$
<>	Inequality	$\$x <> \$y$	Returns true if $\$x$ is not equal to $\$y$
!==	Non-identity	$\$x !== \$y$	Returns true if $\$x$ is not identical to $\$y$

- ▶ What is the output of the following code:

```
<?php
$x = "Hayder";
$y = "Ahmed";
$z = $x.$y;
$x.=$y;
echo $x. " & " . $y . " & " . $z;
?>
```

- Hayder & Ahmed & HayderAhmed
- Hayder & HayderAhmed & HayderAhmed
- HayderAhmed & Ahmed & HayderAhmed

- ▶ What is the output of the following code:

```
<?php
$x = 100;
$y = 100.00;
var_dump($x === $y);
?>
```

- bool(true)
- bool(false)

# Questions



# الجامعة التقنية الجنوبية المعهد التقني القرنة قسم تقنيات أنظمة الحاسوب



اساسيات تصميم المواقع CST106

محاضرة رقم (٢) - HTML



# HTML Text Formatting

- ▶ HTML defines special elements for defining text with a special meaning.

Tag	Definition	Example	Appearance
<b>	Bold text	<b>This text is bold</b>	<b>This text is bold</b>
<strong>	Important text	<strong>This text is strong</strong>	<b>This text is strong</b>
<i>	Italic text	<i>This text is italic</i>	<i>This text is italic</i>
<em>	Emphasized text	<em>This text is emphasized</em>	<i>This text is emphasized</i>
<mark>	Marked text	<mark>Marked</mark>	<b>Marked</b>
<small>	Small text	<small>Small</small>	Small
<del>	Deleted text	My favorite color is <del>blue</del>	My favorite color is <del>blue</del>
<ins>	Inserted text	My favorite <ins>color</ins> is red	My favorite <u>color</u> is red
<sub>	Subscript text	This is <sub>subscripted</sub> text	This is <sub>subscripted</sub> text
<sup>	Superscript text	This is <sup>superscripted</sup> text	This is <sup>superscripted</sup> text

# HTML Links

▶ HTML links are used to link from one page to another (web page, file, website, etc.).

▶ The HTML hyperlink has the following syntax:

```
<a href="url"> link text </a>
```

▶ **href** attribute specifies the destination address of the link.

▶ **link text** is the visible part.

```
<a href="https://www.w3schools.com"> Visit our website</a>
```

External link

```
<a href="/html/contact.html" target="_blank"> Contact Us </a>
```

Local link

▶ The **target** attribute specifies where to open the linked document.

target

=

\_blank

opens in  
new  
window

\_self

opens  
in same  
window

\_parent

opens in  
parent  
frame

\_top

opens in  
full body  
of the  
window

Frame\_name

opens in a  
named frame  
(**name attribute**)

# HTML Link Bookmarks

- ▶ HTML bookmarks are used to allow readers to jump to specific parts of a Web page.
- ▶ To create a bookmark:

1 create the bookmark

```
<h2 id="C4"> Chapter 4 </h2>
```

2 add a link to bookmark

```
<a href="#C4"> Jump to Chapter 4 </a>
```

Same page

```
<a href="page.html#C5"> Jump to Chapter 5 </a>
```

Another page

- ▶ An unvisited link will appear underlined and blue

[Jump to Chapter 4](#)

- ▶ A visited link will appear underlined and purple

[Jump to Chapter 4](#)

- ▶ An active link will appear underlined and red

[Jump to Chapter 4](#)

# HTML Images

- ▶ In HTML, images are defined with the `<img>` tag (empty element).
- ▶ `<img>` tag has the following syntax: ``
- ▶ `alt` attribute provides an alternate text for an image, if the browser cannot find it.

```

```

- ▶ `width` and `height` attributes are used to specify the width and height of an image.

```

```

- ▶ `style` attribute is used to specify the width and height of an image using CSS `height` and `width` properties.

```

```

- ▶ the CSS `float` property is used to let the image float to the `right` or to the `left` of a text.

```

```

What are  
`<map>` tag and  
`usemap`  
attribute?

# HTML Picture Element

- ▶ The picture element allows to display different pictures for different devices or screen sizes.
- ▶ The `<picture>` element contains a number of `<source>` elements.
- ▶ Each `<source>` element has attributes describing when their image is the most suitable.

```
<picture>  
  <source media="(min-width: 650px)" srcset="img1.jpg">  
  <source media="(min-width: 465px)" srcset="img2.jpg">  
    
</picture>
```

- ▶ Always specify `<img>` as the last child of the `<picture>` element. The `<img>` is used by browsers that do not support the `<picture>` element, or if none of the `<source>` tags matched.
- ▶ Two main purposes for the `<picture>` element:
  - ▶ Bandwidth
  - ▶ Format Support



HTML5



# الجامعة التقنية الجنوبية المعهد التقني القرنة قسم تقنيات أنظمة الحاسوب



تصميم المواقع الالكترونية

محاضرة رقم (٣) - PHP

# PHP Conditional Statements

- ▶ Conditional statements are used to perform different actions based on different conditions.
- ▶ **if** statement - executes some code if one condition is true.
- ▶ **if...else** statement - executes some code if a condition is true and another code if that condition is false.
- ▶ **if...elseif...else** statement - executes different codes for more than two conditions.

```
if (condition) {  
    code to be executed if  
    condition is true;  
}
```

```
if (condition) {  
    code to be executed if  
    condition is true;  
} else {  
    code to be executed if  
    condition is false;  
}
```

```
if (condition_1) {  
    code to be executed if  
    condition_1 is true;  
} elseif (condition_2) {  
    code to be executed if  
    condition_2 is true;  
} else {  
    code to be executed if  
    all conditions are false;  
}
```

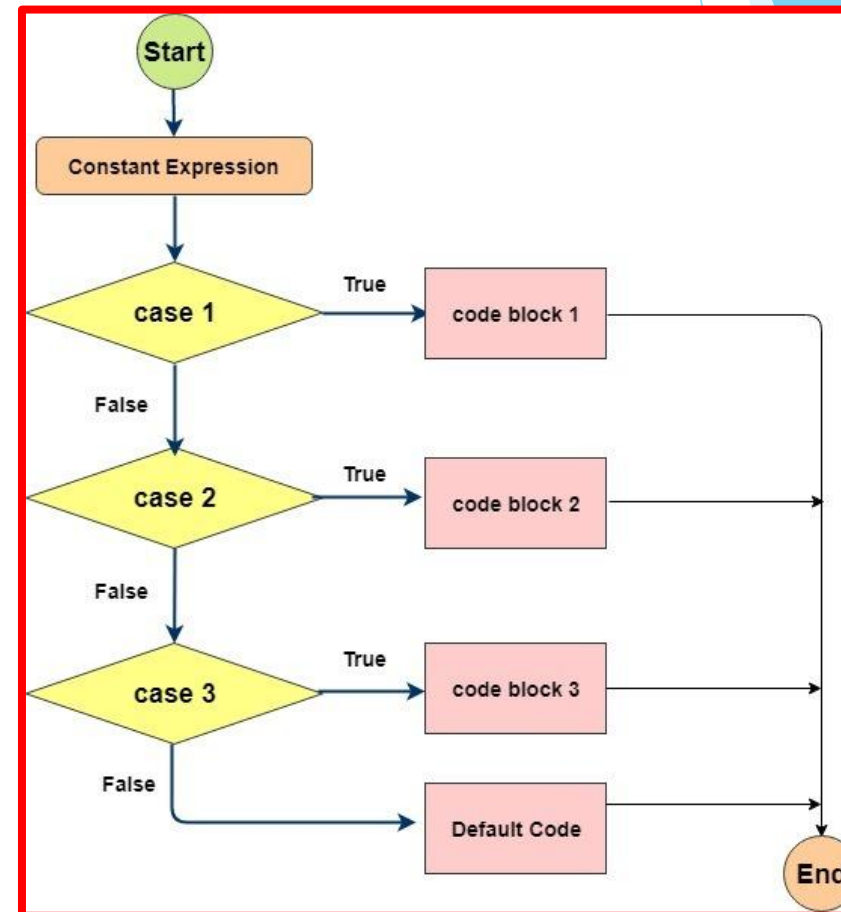
## Homework: Student Grading System

Write a php code to convert a student mark to a grade using if...elseif...else? (Fail, Pass, Medium, Good, Very Good, Excellent).

# PHP switch Statement

- ▶ The **switch** statement is used to perform different actions based on different conditions.
- ▶ Use the **switch** statement to select one of many blocks of code to be executed.

```
switch (n) {  
  case label1:  
    code to be executed if n=label1;  
    break;  
  case label2:  
    code to be executed if n=label2;  
    break;  
  ...  
  default:  
    code to be executed if n is different  
    from all labels;  
}
```





▶ What is the output of the following code:

- \*\*\*\*\*
- \*\*\*\*\* \$\$\$\$\$
- ##### \$\$\$\$\$
- #####
- \$\$\$\$\$

```
<?php
$x = 9;
$y = 11;
if ( $x > 10 )
if ( $y > 10 )
echo "***** ";
else
echo "##### ";
echo "$$$$$ ";
?>
```

▶ What is the output of the following code:

- number 1
- number 1    number 2
- number 2
- number 1    another number
- number 2    another number
- another number
- error

```
<?php
$n = 1;
switch($n)
{
case 1:
echo "number 1    ";
case 2:
echo "number 2    ";
break;
default:
echo "number is not 1 or 2";
}
?>
```

# Questions

# PHP Loops: While

- ▶ Loops are used to execute the same block of code again and again, as long as a certain condition is true.
- ▶ The **while** loop executes a block of code as long as the specified condition is true.

```
while (condition is true) {  
    code to be executed;  
}
```

```
<?php  
$x = 0;  
while($x <= 100) {  
    $x+=20;  
    echo "The number is: $x <br>";  
}  
?>
```

```
The number is: 20  
The number is: 40  
The number is: 60  
The number is: 80  
The number is: 100  
The number is: 120
```

Homework:  
Factorial Calculation

Write a php code to calculate the factorial of 5 using while loop statement?

# PHP Loops: Do...While

- ▶ The **do...while** loop will always execute the block of code once, it will then check the condition, and repeat the loop while the specified condition is true.

```
do {  
    code to be executed;  
} while (condition is true);
```

```
<?php  
$x = 0;  
do {  
    $x+=20;  
    echo "The number is: $x <br>";  
} while($x <= 100);  
?>
```

```
The number is: 20  
The number is: 40  
The number is: 60  
The number is: 80  
The number is: 100  
The number is: 120
```

Homework:  
Factorial Calculation

Write a php code to calculate the factorial of 5 using do...while loop statement?

# PHP Loops: For

- ▶ The **for** loop is used when you know in advance how many times the script should run.

```
for (initial counter; test counter; increment counter) {  
    code to be executed for each iteration;  
}
```

What is  
**foreach** loop ?

```
<?php  
for ($x = 0; $x <= 100; $x+=20) {  
    echo "The number is: $x <br>";  
}  
?>
```

```
The number is: 0  
The number is: 20  
The number is: 40  
The number is: 60  
The number is: 80  
The number is: 100
```

Homework:  
Factorial Calculation

Write a php code to calculate the factorial of 5 using for loop statement?

▶ What is the output of the following code:

- inside while
- inside while  
program terminated
- program terminated
- error

```
<?php
$n = 10;
while($n<10)
{
echo "inside while <br>";
$n++;
}
echo "program terminated"
?>
```

▶ What is the output of the following code:

- inside while
- inside while  
program terminated
- program terminated
- error

```
<?php
$n = 10;
do
{
echo "inside while <br>";
$n++;
} while($n<10);
echo "program terminated"
?>
```

# Questions



# الجامعة التقنية الجنوبية المعهد التقني القرنة قسم تقنيات أنظمة الحاسوب

اساسيات تصميم المواقع CST106

محاضرة رقم (٣) - HTML

# HTML Tables

- ▶ An HTML table is defined with the `<table>` tag.
- ▶ Each table row is defined with the `<tr>` tag.
- ▶ A table header is defined with the `<th>` tag.
- ▶ A table data/cell is defined with the `<td>` tag.
- ▶ The `colspan` attribute is used to merge cells horizontally.

```
<td colspan="2"> Hayder </td>
```

- ▶ The `rowspan` attribute is used to merge cells vertically.

```
<td rowspan="2"> Hayder </td>
```

- ▶ The `<caption>` tag is used to add a caption to the table.
- ▶ The `<caption>` must be inserted immediately after `<table>`.

```
<caption> Table Title </caption>
```

First name	Last name
Hayder	Ahmed
Hassan	Ali

```
<table style="width:100%">
  <tr>
    <th> First name </th>
    <th> Last name </th>
  </tr>
  <tr>
    <td> Hayder </td>
    <td> Ahmed </td>
  </tr>
  <tr>
    <td> Hassan </td>
    <td> Ali </td>
  </tr>
</table>
```

# CSS Table Properties

Property	Description	Value
background-color	sets background color	blue, green, red,...
color	sets text color	blue, green, red,...
width	defines width of a table	50%, 100%, 50px, 100px,...
height	defines height of a table.	50%, 100%, 50px, 100px,...
text-align	aligns cell text horizontally	left, right, or center
vertical-align	aligns cell text vertically	top, bottom, or middle
border-spacing	specifies the space between the cells	5px, 10px,...
padding	specifies the space between the cell content and its borders	5px, 10px, 15px,...
border	sets a table border	1px solid black
border-collapse	collapses the borders into one border	collapse, separate



# HTML Tables

```
<!DOCTYPE html>
<html>
<head>
<style>
Table {
width: 50%;
height: 50%;
border: 1px solid black;
color: white;
background-color: #33A8FF;
border-spacing: 15px;
font-size: xx-large;
}
th, td {
border: 1px solid black;
padding: 15px;
text-align: center;
vertical-align: middle;
}
</style>
</head>
```

```
<body>
<table>
<tr> <th>Firstname</th>
<th>Lastname</th>
<th>Age</th>
</tr>
<tr> <td>Hayder</td>
<td>Ahmed</td>
<td>Abdulmosin</td>
</tr>
<tr> <td>Hassan</td>
<td>Ali</td>
<td>Mohammed</td>
</tr>
<tr> <td>Waleed</td>
<td>Rawad</td>
<td>Ehsan</td>
</tr>
</table>
</body>
</html>
```

Firstname	Lastname	Age
Hayder	Ahmed	Abdulmosin
Hassan	Ali	Mohammed
Waleed	Rawad	Ehsan

# HTML Lists

- ▶ Two types: Unordered List and Ordered List.
- ▶ An unordered list starts with the `<ul>` tag.
- ▶ An ordered list starts with the `<ol>` tag.
- ▶ Each list item starts with the `<li>` tag.
- ▶ The CSS `list-style-type` property is used to define the style of the list item marker in `<ul>` tag.

list-style-type



disc

square

circle

none

- ▶ The `type` attribute of the `<ol>` tag, defines the type of the list item marker.

type



1

A

a

l

i

- ▶ the `start` attribute is used to start counting from a specified number in `<ol>` tag.
- ▶ The `<dl>` tag defines the description list, the `<dt>` tag defines the term (name), and the `<dd>` tag describes each term:

# HTML Lists

```
<!DOCTYPE html>
<html>
<body>
<ul style="list-style-type:disc;">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ul>
</body>
</html>
```

- Coffee
- Tea
- Milk

```
<!DOCTYPE html>
<html>
<body>
<ol type="1" start="11">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ol>
</body>
</html>
```

11. Coffee
12. Tea
13. Milk

```
<!DOCTYPE html>
<html>
<body>
<dl>
  <dt>Coffee</dt>
  <dd>- black hot drink</dd>
  <dt>Milk</dt>
  <dd>- white cold drink</dd>
</dl>
</body>
</html>
```

Coffee  
- black hot drink

Milk  
- white cold drink



# الجامعة التقنية الجنوبية المعهد التقني القرنة قسم تقنيات أنظمة الحاسوب



تصميم المواقع الالكترونية

محاضرة رقم (٤) - PHP

# PHP Functions

- ▶ PHP has more than 1000 built-in functions.
- ▶ A user-defined function declaration starts with the word **function**:

```
function functionName() {  
    code to be executed;  
}
```

```
<?php  
function writeMsg() {  
    echo "Hello world!";  
}  
writeMsg();  
?>
```

- ▶ Information can be passed to functions through **arguments**.

```
<?php  
function writeMsg($x) {  
    echo $x;  
}  
writeMsg("Hello STU!");  
?>
```

```
<?php  
function printName($name, $year) {  
    echo "$name was born in $year";  
}  
printName("Hayder", 1986);  
?>
```

Write a  
function with  
**THREE**  
arguments?

## PHP Functions (Contin.)

- ▶ PHP automatically associates a data type to the variable, depending on its value.
- ▶ In PHP 7, type declarations were added. This gives us an option to specify the expected data type when declaring a function, and by adding the **strict** declaration (**strict mode**).
- ▶ To specify strict (strict mode) we need to set **declare(strict\_types=1);** in the first line of the PHP file.

```
<?php  
function addNumbers(int $a, int $b) {  
    return $a + $b;  
}  
echo addNumbers(5, "5 days");  
?>
```

10

```
<?php declare(strict_types=1);  
function addNumbers(int $a, int $b) {  
    return $a + $b;  
}  
echo addNumbers(5, "5 days");  
?>
```

error

### Homework:

### Function with Strict Mode

Write a php code to sum two float numbers using function with enabling strict mode?

▶ What is the output of the following code:

- 5
- 10
- 5 hayder
- 10 hayder
- error

```
<?php
function add(int $a, int $b) {
    return $a + $b;
}
echo add(5, "hayder 5");
?>
```

▶ What is the output of the following code:

- 5
- 10
- 55
- error

```
<?php declare(strict_types=1);
function add(int $a, int $b) {
    return $a + $b;
}
echo add(5, "5");
?>
```

# Questions

## PHP Functions (Contin.)

- ▶ If we call a function without arguments, it takes the default value as argument.
- ▶ PHP 7 also supports Type Declarations for the return statement.

```
<?php declare(strict_types=1);  
function setHeight(int $height = 50) {  
    echo "The height is : $height <br>";  
}  
setHeight(350);  
setHeight();  
?>
```

350  
50

```
<?php declare(strict_types=1);  
function addNumbers(float $a, float $b) : float {  
    return $a + $b;  
}  
echo addNumbers(1.2, 5.2);  
?>
```

6.4

### Homework:

### Function with Strict Mode

Write a php code to sum two float numbers (x and y) using function where the default values of x = 10 and y = 20?



▶ What is the output of the following code:

- 35.5 # 50 # 25 #
- 35.5 # 25 #
- 50 # 25 #
- 35 # 50 # 25 #
- error

```
|k?php declare(strict_types=1);  
function set(int $height = 50) {  
    echo "$height # ";  
}  
set(35.5);  
set();  
set(25);  
?>
```

▶ What is the output of the following code:

- 150
- 75
- 125
- 50
- 100
- error

```
<?php declare(strict_types=1);  
function sum(int $x = 50, $y = 100) {  
    return $x + $y;  
}  
echo sum(25);  
?>
```

# Questions

▶ What is the output of the following code:

- 6.2
- 6
- 1
- 5
- error

```
<?php declare(strict_types=1);  
function add(int $a, float $b) : int {  
    return $a + $b;  
}  
echo add(1, 5.2);  
?>
```

▶ What is the output of the following code:

- 11.5
- 11
- 6.3
- 5.2
- error

```
<?php declare(strict_types=1);  
function add(float $a, float $b) : int {  
    return (int)($a + $b);  
}  
echo add(6.3, 5.2);  
?>
```

# Questions



# الجامعة التقنية الجنوبية المعهد التقني القرنة قسم تقنيات أنظمة الحاسوب

اساسيات تصميم المواقع CST106

محاضرة رقم (٤) - HTML

# HTML Iframes

- ▶ An iframe is used to display a web page within a web page.
- ▶ An HTML iframe is defined with the `<iframe>` tag:

```
<iframe src="URL"> </iframe>
```

- ▶ The height and width attributes are used to specify the size of the iframe.

```
<iframe src="https://stu.edu.iq" height="200" width="300"> </iframe>
```

- ▶ Also, the CSS height and width properties are used to specify the size of the iframe.

```
<iframe src="https://stu.edu.iq" style="height:200px; width:300px;"> </iframe>
```

- ▶ The CSS border property is used to change the size, style and color of the iframe's border

```
<iframe src="https://stu.edu.iq" style="border:2px solid red;"> </iframe>
```

# HTML Video

- ▶ The HTML5 **<video>** element specifies a standard way to embed a video in a web page.

```
<video width="320" height="240" controls>
  <source src="movie.mp4" type="video/mp4">
  <source src="movie.ogg" type="video/ogg">
  Your browser does not support the video tag.
</video>
```

```
<video width="320" height="240" autoplay>
  <source src="movie.mp4" type="video/mp4">
  <source src="movie.ogg" type="video/ogg">
  Your browser does not support the video tag.
</video>
```

- ▶ The **controls** attribute adds video controls, like play, pause, and volume.
- ▶ The **<source>** element allows you to specify alternative video files which the browser may choose from.
- ▶ The text between the **<video>** element will be displayed in browsers that do not support the **<video>**.
- ▶ The **autoplay** attribute is used to start a video automatically.

Element					
<video>	4.0	9.0	3.5	4.0	10.5

# HTML Audio

- ▶ The HTML5 **<audio>** element specifies a standard way to embed audio in a web page.

```
<audio controls>
  <source src="horse.ogg" type="audio/ogg">
  <source src="horse.mp3" type="audio/mpeg">
  Browser does not support the audio element.
</audio>
```

```
<audio controls>
  <source src="horse.ogg" type="audio/ogg">
  <source src="horse.mp3" type="audio/mpeg">
  Browser does not support the audio element.
</audio>
```

- ▶ The **controls** attribute adds audio controls, like play, pause, and volume.
- ▶ The **<source>** element allows you to specify alternative audio files which the browser may choose from.
- ▶ The text between the **<audio>** element will be displayed in browsers that do not support the **<audio>**.
- ▶ The **autoplay** attribute is used to start audio automatically.

Element					
<video>	4.0	9.0	3.5	4.0	10.5

# HTML YouTube Videos

- ▶ The easiest way to play videos in HTML, is to use YouTube.
- ▶ To play your video on a web page, do the following:
  - A. Upload the video to YouTube.
  - B. Take a note of the **video id**.
  - C. Define an **<iframe>** element in your web page
  - D. Let the **src** attribute point to the video URL
  - E. Use the **width** and **height** attributes to specify the dimension of the player
  - F. Add any other parameters to the URL (e.g. **autoplay**, **loop**, **controls**).

```
<iframe width="420" height="315"
src="https://www.youtube.com/embed/5yJW8zLtbfs">
</iframe>
```

```
<iframe width="420" height="315" src="https://www.youtube.com/embed/5yJW8zLtbfs?autoplay=1">
</iframe>
```

```
<iframe width="420" height="315" src="https://www.youtube.com/embed/5yJW8zLtbfs?loop=1">
</iframe>
```

# HTML Forms

- ▶ The HTML **<form>** element defines a form that is used to collect user input.
- ▶ An HTML form contains different types of input elements e.g. text fields, checkboxes, radio buttons, submit buttons, and more.
- ▶ The **<input>** element is the most important form element.
- ▶ The **<input>** element is displayed in several ways, depending on the **type** attribute.
- ▶ The **<label>** tag defines a label for many form elements.

```
<body>
<form>
  <label for="fname">First name:</label><br>
  <input type="text" id="fname" name="fname" value="Hayder"><br>
  <label for="lname">Last name:</label><br>
  <input type="text" id="lname" name="lname" value="Ahmed">
</form>
</body>
```

First name:

Last name:





# الجامعة التقنية الجنوبية المعهد التقني القرنة قسم تقنيات أنظمة الحاسوب



تصميم المواقع الالكترونية

محاضرة رقم (٥) - PHP

# PHP Arrays

- ▶ An array stores multiple values in one single variable.
- ▶ In PHP, the **array()** function is used to create an array:

```
<?php
$x = array(2, 4, 6);
echo " Elements of x = " . $x[0] . ", " . $x[1] . " and " . $x[2] . ".";
?>
```

- ▶ The **count()** function is used to return the length (the number of elements) of an array.

```
<?php
$x = array(2, 4, 6);
echo " Number of elements in x = " . count($x);
?>
```

- ▶ In PHP, three types of arrays are: **Indexed arrays** - Arrays with a numeric index, **Associative arrays** - Arrays with named keys, **Multidimensional arrays** - Arrays containing one or more arrays

# PHP Indexed Arrays

- ▶ Indexed arrays - Arrays with a numeric index.
- ▶ Two ways to create indexed arrays:
- ▶ The index can be assigned automatically (index starts at 0).
- ▶ The index can be assigned manually.

```
$x = array(2, 4, 6);
```

```
$x[0] = 2; $x[1] = 4; $x[2] = 6;
```

```
<?php
$family = array("Hayder", "Ali", "Hassan", "Mohammed");
$length = count($family);
for($i = 0; $i < $length; $i++) {
    echo $family[$i];
    echo "<br>";
}
?>
```

Create an array containing your family members, then print the array elements?

▶ What is the output of the following code:

- 2 4 6 8 10
- 2 4 6
- 2 6 10
- 4 8
- error

```
<?php
$x = array(2, 4, 6, 8, 10);
$length = count($x);
for($i = 0; $i < $length; $i=$i+2) {
    echo $x[$i] . " ";
}
?>
```

▶ What is the output of the following code:

- 120
- 15
- 12
- 0
- error

```
<?php
$x = array(1, 2, 3, 4, 5);
$p = 0;
$length = count($x);
for($i = 0; $i < $length; $i++) {
    $p = $p*$x[i];
}
echo $p;
?>
```

# Questions

# PHP Associative Arrays

- ▶ Associative arrays - Arrays with named keys.
- ▶ Two ways to create an associative array:

```
$age = array("Ali"=>"35", "Ahmed"=>"37", "Zain"=>"43");
```

```
$age['Ali'] = "35"; $age['Ahmed'] = "37"; $age['Zain'] = "43";
```

```
<?php
$age = array("Ali"=>"35", "Ahmed"=>"37", "Zain"=>"43");
$keys = array_keys($age);
for($i=0; $i<count($keys); $i++) {
    echo "Key=" . $keys[$i] . ", Value=" . $age[$keys[$i]];
    echo "<br>";
}
?>
```

Ali  
Ahmed  
Zain } Keys

35  
37  
43 } Values

Rewrite the example on the left using **foreach** statement instead of **for** statement?

# Sorting Arrays

- ▶ The elements in an array can be sorted in alphabetical or numerical order, descending or ascending using PHP array sort functions.
- ▶ **sort()** - sort arrays in ascending order.
- ▶ **rsort()** - sort arrays in descending order.
- ▶ **asort()** - sort associative arrays in ascending order, according to the value.
- ▶ **ksort()** - sort associative arrays in ascending order, according to the key.
- ▶ **arsort()** - sort associative arrays in descending order, according to the value.
- ▶ **krsort()** - sort associative arrays in descending order, according to the key.

```
<?php
$x = array(4, 6, 2, 22, 11);
sort($x);
for($i = 0; $i < count($x); $i++) {
    echo $x[$i] . " "; }
?>
```

Sort the associative array,  
in example on page 37, in  
descending order  
according to the value?

▶ What is the output of the following code:

- B , A
- C , B
- 20 , 10
- 30 , 20
- error

```
<?php
$x = array("A"=>"10", "B"=>"20", "C"=>"30");
$keys = array_keys($x);
echo $keys[2] . " , " . $keys[1];
?>
```

▶ What is the output of the following code:

- Hassan
- Ali
- Zain
- 35
- 37
- 43
- error

```
<?php
$age = array("Hassan"=>"35", "Zain"=>"37", "Ali"=>"43");
krsort($age);
$key = array_keys($age);
echo $key[0];
?>
```

# Questions



# الجامعة التقنية الجنوبية المعهد التقني القرنة قسم تقنيات أنظمة الحاسوب



اساسيات تصميم المواقع CST106

محاضرة رقم (٥) - HTML



# HTML Form Elements

- ▶ The **<select>** element defines a drop-down list.
- ▶ The **<option>** elements defines an option that can be selected.
- ▶ The **selected** attribute defines a pre-selected option.
- ▶ The **size** attribute specifies the number of visible values.
- ▶ the **multiple** attribute allows the user to select multiple values.
- ▶ The **<textarea>** element defines a multi-line input field.

```
<textarea name="message" rows="10" cols="30">
The cat was playing in the garden. </textarea>
```

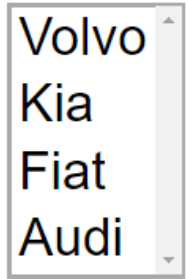
- ▶ The **<button>** element defines a clickable button.

```
<button type="button" onclick="alert('Hello World!')">
Click Me! </button>
```

- ▶ The **<fieldset>** element is used to group related data in a form.
- ▶ The **<legend>** element defines a caption for the **<fieldset>** element.

```
<form>
<label for="cars">Choose a car:</label>
<select id="cars" name="cars" size="4">
<option value="Volvo">Volvo</option>
<option value="Kia">Kia</option>
<option value="fiat">Fiat</option>
<option value="audi">Audi</option>
</select>
</form>
```

Choose a car:



# HTML Input Types

Input Type	Description
text	defines a single-line text input field
password	defines a password field
submit	defines a button for submitting data
radio	defines a radio button
checkbox	defines a checkbox
color	shows a color picker
date	shows a date picker
file	defines a file-select field
number	defines a numeric input field
button	defines a button

## Radio Buttons

- Male
- Female
- Other

## Checkboxes

- I have a bike
- I have a car
- I have a boat

# HTML input Types

```
<form action="/action_page.php">
  <label for="favcolor">Select your favorite color:</label>
  <input type="color" id="favcolor" name="favcolor" value="#ff0000">
  <input type="submit" value="Submit">
</form>
```

```
<form action="/action_page.php">
  <label for="birthday">Birthday:</label>
  <input type="date" id="birthday" name="birthday">
  <input type="submit" value="Submit">
</form>
```

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

<h1>File upload</h1>

<p>Show a file-select field which allows a file to be chosen for upload:</p>
<form action="/action_page.php">
  <label for="myfile">Select a file:</label>
  <input type="file" id="myfile" name="myfile"><br><br>
  <input type="submit" value="Submit">
</form>

</body>
</html>
```

## Show a Color Picker

The `input type="color"` is used for input fields that should contain a color.

Select your favorite color:

**Note:** `type="color"` is not supported in Internet Explorer 11 or Safari 9.1 (or earlier).

## File upload

Show a file-select field which allows a file to be chosen for upload:

Select a file:

# The input type

When the Run

```
<form action="/action_page.php">  
  <label for="fname">First name:</label><br>  
  <input type="text" id="fname" name="fname" value="johan"><br>  
  <label for="lname">Last name:</label><br>  
  <input type="text" id="lname" name="lname" value="doc"><br><br>  
  <input type="submit" value="Submit">  
  <input type="reset">  
</form>
```

First name:

johan

Last name:

doc

Submit

Reset

If you change the input values and then click the "Reset" button, the form-data will be reset to the default values.

First name:

Last name:

Submit

Reset

When we put the value "" the default no thing

# HTML Methods

- ▶ GET
- ▶ POST
- ▶ PUT
- ▶ HEAD
- ▶ DELET
- ▶ PATCH
- ▶ OPTIONS
- ▶ THE two common HTTP methods are :GET and POST

# HTML Methods

## The GET Method

GET is used to request data from a specified resource.

GET is one of the most common HTTP methods.

Note that the query string (name/value pairs) is sent in the URL of a GET request:

---

```
/test/demo_form.php?name1=value1&name2=value2
```

---

## The POST Method

POST is used to send data to a server to create/update a resource.

The data sent to the server with POST is stored in the request body of the HTTP request:

```
POST /test/demo_form.php HTTP/1.1
```

```
Host: w3schools.com
```

```
name1=value1&name2=value2
```



# GET

# POST

BACK button/Reload	Harmless	Data will be re-submitted (the browser should alert the user that the data are about to be re-submitted)
Bookmarked	Can be bookmarked	Cannot be bookmarked
Cached	Can be cached	Not cached
Encoding type	application/x-www-form-urlencoded	application/x-www-form-urlencoded or multipart/form-data. Use multipart encoding for binary data
History	Parameters remain in browser history	Parameters are not saved in browser history
Restrictions on data length	Yes, when sending data, the GET method adds the data to the URL; and the length of a URL is limited (maximum URL length is 2048 characters)	No restrictions
Restrictions on data type	Only ASCII characters allowed	No restrictions. Binary data is also allowed
Security	GET is less secure compared to POST because data sent is part of the URL  Never use GET when sending passwords or other sensitive information!	POST is a little safer than GET because the parameters are not stored in browser history or in web server logs
Visibility	Data is visible to everyone in the URL	Data is not displayed in the URL

# Example

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

<h2>Password field</h2>

<p>The <strong>input type="password"</strong> defines a password field:</p>

<form action="/action_page.php">
  <label for="username">Username:</label><br>
  <input type="text" id="username" name="username"><br>
  <label for="pwd">Password:</label><br>
  <input type="password" id="pwd" name="pwd"><br><br>
  <input type="submit" value="Submit">
</form>

<p>The characters in a password field are masked (shown as asterisks or
circles).</p>

</body>
</html>
```

## Password field

The `input type="password"` defines a password field:

Username:

Password:

The characters in a password field are masked (shown as asterisks or circles).