WEB PROGRAMMING- PHP

WEEK 1

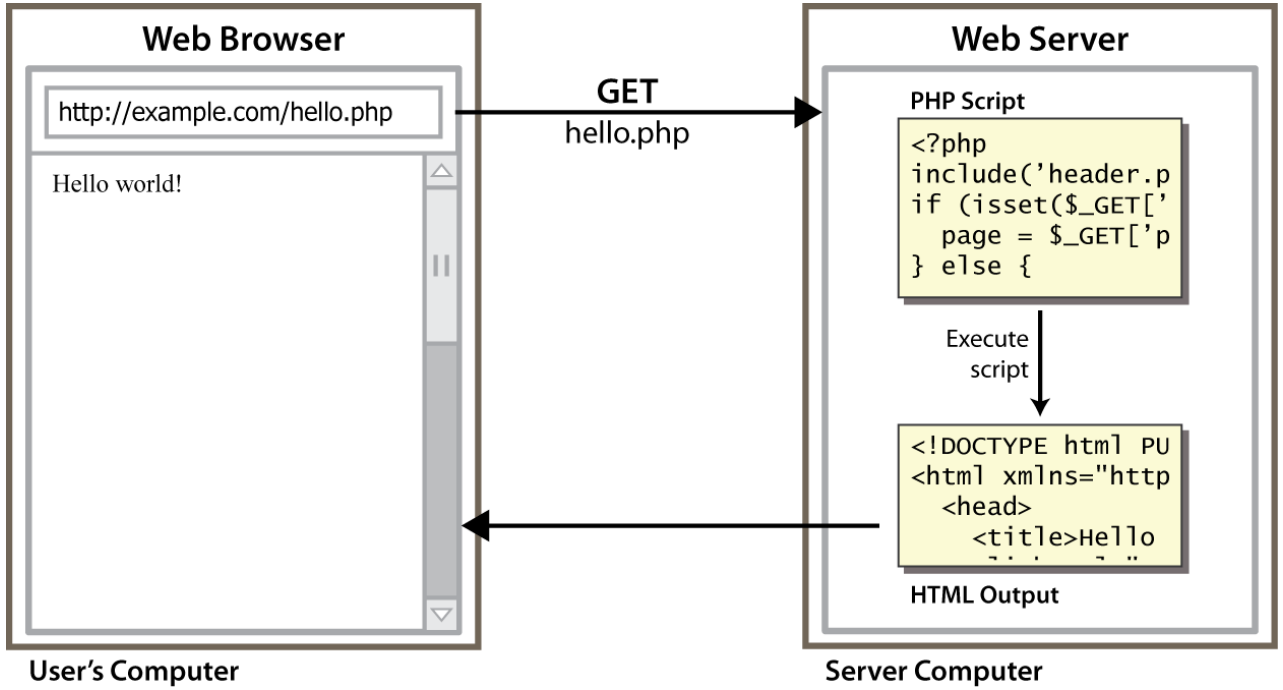
WHAT IS PHP?

What is PHP?

- PHP is an acronym for "PHP: Hypertext Preprocessor"
- PHP is a widely-used, scripting language
- PHP scripts are executed on the server
- PHP is free to download and use

- It is also easy enough to be a beginner's first server side language!

LIFECYCLE OF A PHP WEB REQUEST



- browser requests a `.html` file (**static content**): server just sends that file
- browser requests a `.php` file (**dynamic content**): server reads it, runs any script code inside it, then

PHP

- Server side scripting language

- Script
    - Run in response to event
    - Not run until webpage is requested
    - Perform sequence of instruction

- Program
    - Run even not respond to event
    - Continuous run waiting for interaction
    - Photoshop, Maya

WHERE IT WORKS

- Server side scripting language
    - Run in the server send result to client
    - Cannot running alone need running web server

- Where Javascript Client Side
    - Run in Client browser

PHP DOES NOT

- Does not need to be compiled
    - Execute by web server exactly as it written

- Java, c++, c# need to compiled to another format to be read

STRUCTURE

- HTML is the structure
    - Php is the content management
- Extension is .php
- Static pages all see same page
- Could create Dynamic data change by user interactions or store in database.

BENEFITS

- **Open source**
- **Cross platform, works on Windows, linus, etc without framework install**
- **Powerful and is an amazing and popular language!**
    - It is powerful enough to be at the core of the biggest blogging system on the web (WordPress)!
    - It is deep enough to run the largest social network (Facebook)!
    - **Used by Facebook, Yahoo, Wikipidia, Wordpress, Flickr**

WE WILL DO

- Learn all PHP basic
- Connect to MYSQL
- Build interaction website

YOU NEED TO KNOW

- XHTML, CSS

THE END

INTRODUCTION TO WEB PROGRAMMING

HTML

WHAT IS HTML?

- HTML is the standard markup language for creating Web pages.
- HTML stands for Hyper Text Markup Language
- HTML describes the structure of Web pages using markup
- HTML elements are the building blocks of HTML pages
- HTML elements are represented by tags
- HTML tags label pieces of content such as "heading", "paragraph", "table", and so on
- Browsers do not display the HTML tags, but use them to render the content of the page

A SIMPLE HTML DOCUMENT

```
<!DOCTYPE html>
<html>
	<head>
		<title>Page Title</title>
	</head>
	<body>
		<h1>My First Heading</h1>
		<p>My first paragraph.</p>
	</body>
</html>
```

EXAMPLE EXPLAINED

- The <!DOCTYPE html> declaration defines this document to be HTML5
- The <html> element is the root element of an HTML page
- The <head> element contains meta information about the document
- The <title> element specifies a title for the document
- The <body> element contains the visible page content
- The <h1> element defines a large heading
- The <p> element defines a paragraph

HTML TAGS

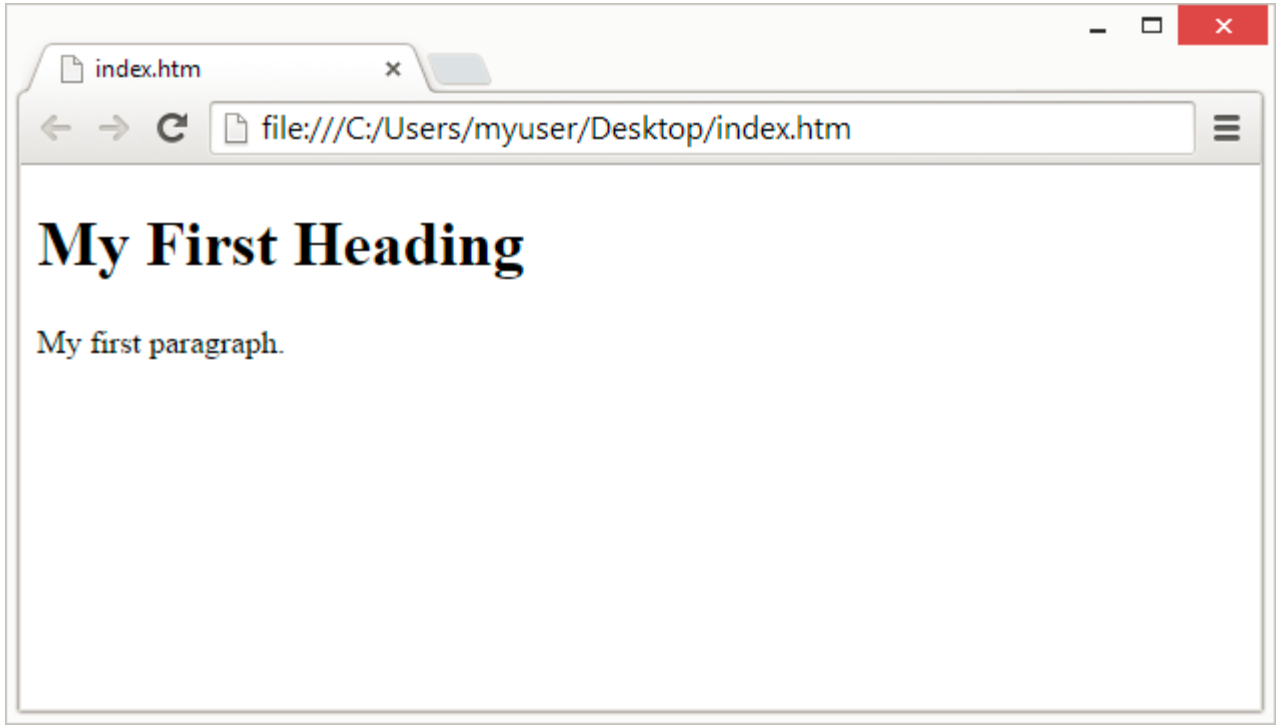
- HTML tags are element names surrounded by angle brackets:

```
<tagname>content goes here...</tagname>
```

- HTML tags normally come **in pairs** like and
- The first tag in a pair is the **start tag,** the second tag is the **end tag**
- The end tag is written like the start tag, but with a **forward slash** inserted before the tag name
- **Tip:** The start tag is also called the **opening tag**, and the end tag the **closing tag**.

WEB BROWSERS

- The purpose of a web browser (Chrome, IE, Firefox, Safari) is to read HTML documents and display them.
- The browser does not display the HTML tags, but uses them to determine how to display the document:



HTML PAGE STRUCTURE

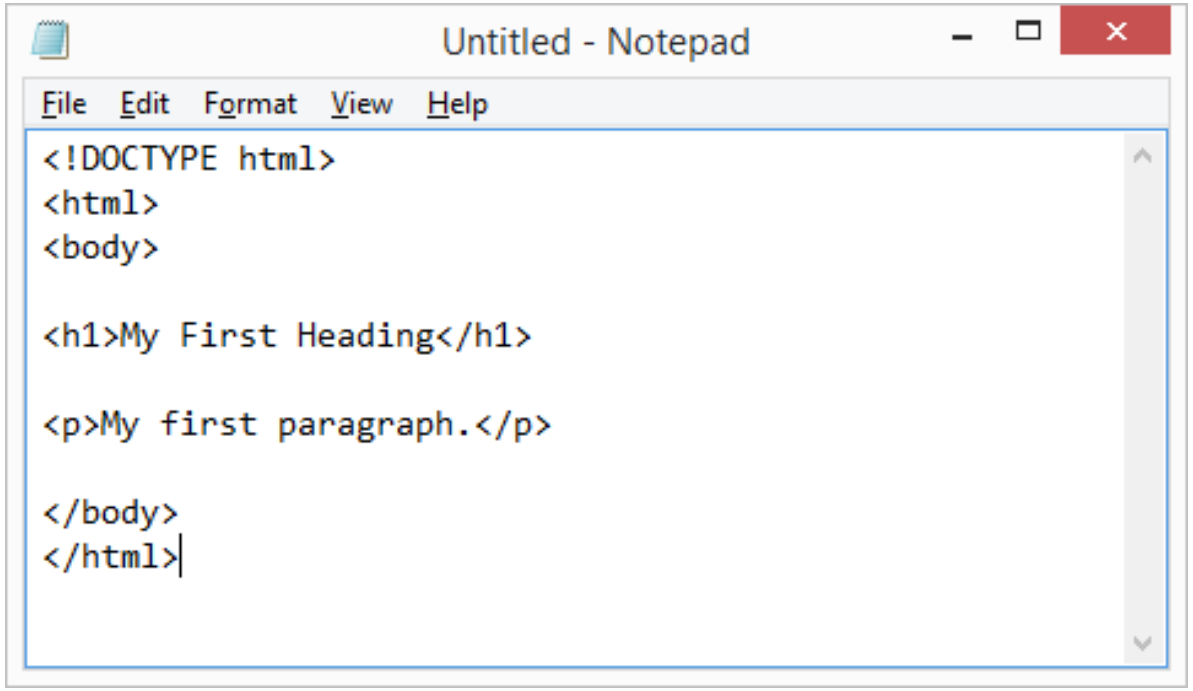
```
<html>
  <head>
    <title>Page title</title>
  </head>
  <body>
    <h1>This is a heading</h1>
    <p>This is a paragraph.</p>
    <p>This is another paragraph.</p>
  </body>
</html>
```

Only the content inside the <body> section (the white area above) is displayed in a browser.

STEPS TO CREATE A WEB PAGE

Follow the four steps below to create your first web page with Notepad/Notepad++ or TextEdit.

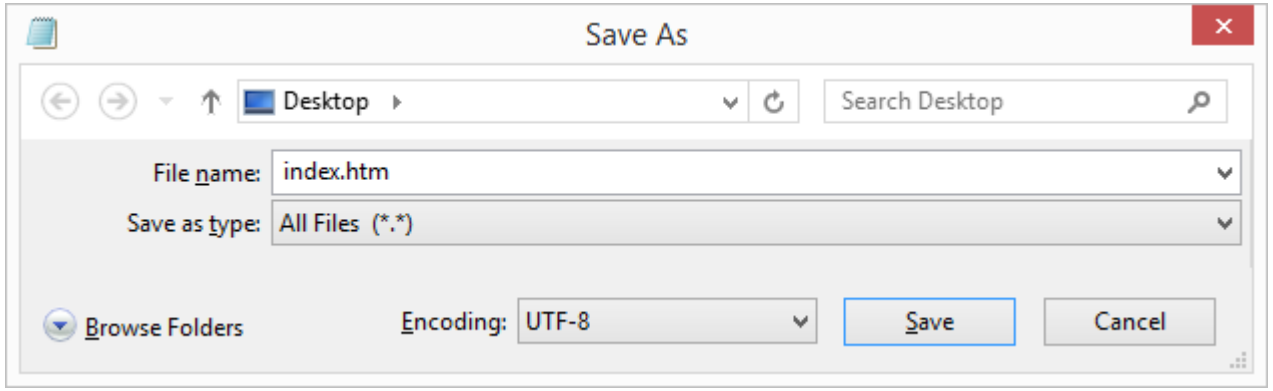
- Step 1: Open Notepad/Notepad++ or TextEdit
- Step 2: Write Some HTML into Notepad/Notepad++ or TextEdit



STEPS TO CREATE A WEB PAGE

Follow the four steps below to create your first web page with Notepad/Notepad++ or TextEdit.

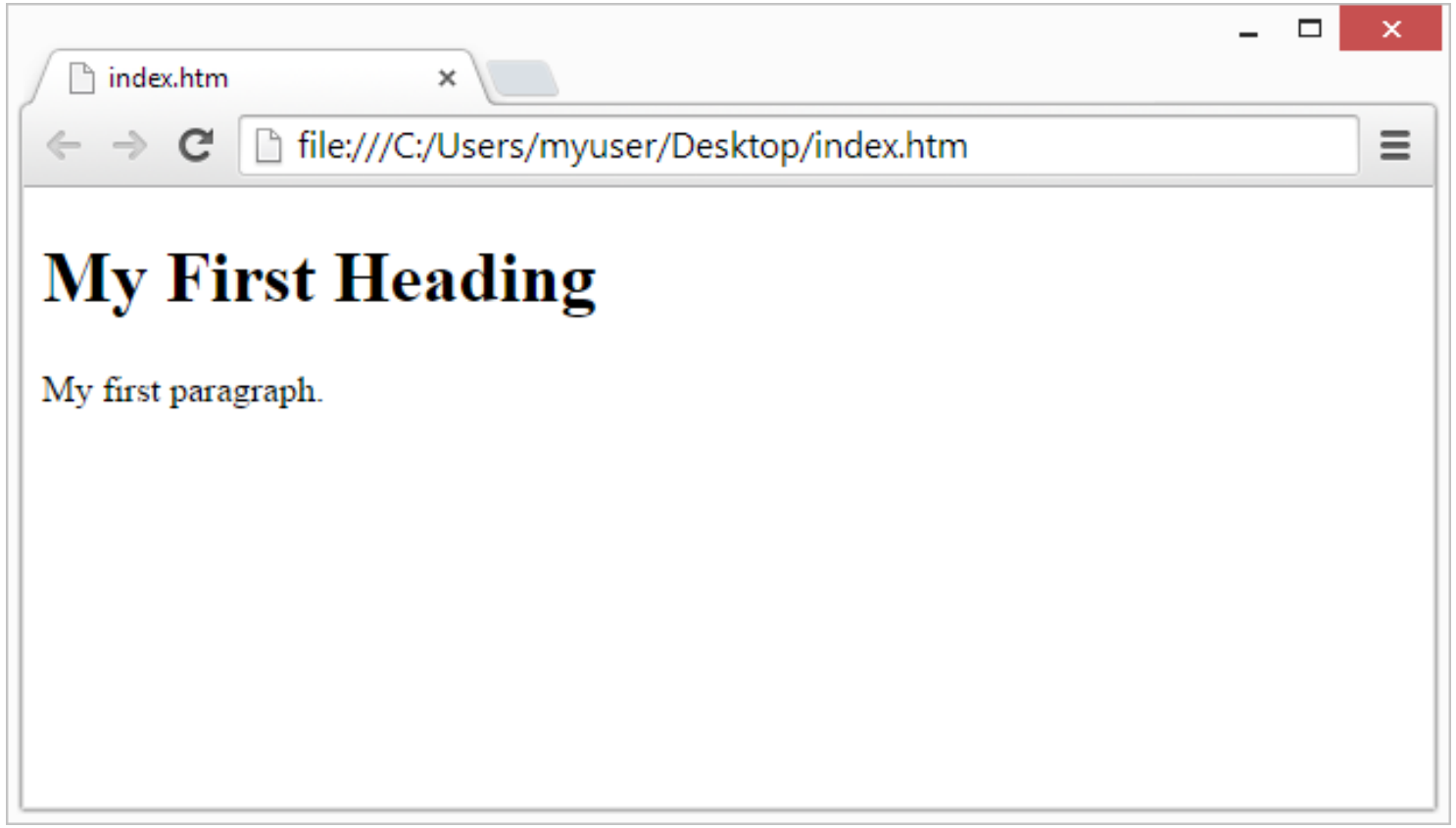
- Step 3: Save the HTML Page



STEPS TO CREATE A WEB PAGE

Follow the four steps below to create your first web page with Notepad or TextEdit.

- Step 4: View the HTML Page in Your Browser



HTML TAGS

HTML DOCUMENTS

- All HTML documents must start with a document type declaration: **<!DOCTYPE html>**.
- The HTML document itself begins with **<html>** and ends with **</html>**.
- The visible part of the HTML document is between **<body>** and **</body>**.

```
<!DOCTYPE html>

<html>
	<body>
		<h1>My First Heading</h1>
		<p>My first paragraph.</p>
	</body>
</html>
```

HTML BASIC TAGS

HTML Headings

- HTML headings are defined with the **<h1>** to **<h6>** tags.
- <h1> defines the most important heading. <h6> defines the least important heading.

```
<h1>This is heading 1</h1>
<h2>This is heading 2</h2>
<h3>This is heading 3</h3>
<h4>This is heading 4</h4>
<h5>This is heading 5</h5>
<h6>This is heading 6</h6>
```

HTML BASIC TAGS

HTML Paragraphs

- HTML paragraphs are defined with the **<p>** tag.

```
<p>This is a paragraph.</p>
<p>This is another paragraph.</p>
```

HTML BASIC TAGS

HTML Line Break

- HTML Line Break is defined with the **
** tag
- HTML elements with no content are called empty elements. Empty elements do not have an end tag such as the Line Break element

```
<br/>
```

HTML Links

- HTML links are defined with the **<a>** tag
- The link's destination is specified in the **href attribute**.
- Attributes are used to provide additional information about HTML elements.

```
<a href="https://www.google.com">This is a link</a>
<a href="page1.html">This is a link</a>
<a href="pages/page1.html">This is a link</a>
```

HTML BASIC TAGS

HTML Images

- HTML images are defined with the **<img>** tag.
- The source file (src), alternative text (alt), width, and height are provided as attributes.
- The **alt** attribute specifies an alternative text to be used, when an image cannot be displayed.

```
<img src="image.jpg" alt="This is an image" width="104" height="142">
<img src="images/image.jpg" alt="This is an image" width="104" height="142">
```

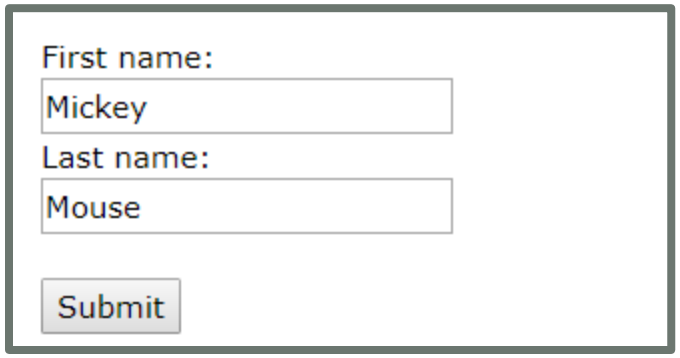
NESTED HTML ELEMENTS

- HTML elements can be nested (elements can contain elements).
- All HTML documents consist of nested HTML elements.

```
<a href="https://www.google.com">
        <img src="image.jpg" alt="google.com" width="104" height="142">
</a>
```

HTML FORM

HTML FORM EXAMPLE




```
<form action="action_page.php">
  First name:<br>
  <input type="text" name="firstname" value="Mickey">
  <br>
  Last name:<br>
  <input type="text" name="lastname" value="Mouse">
  <br><br>
  <input type="submit" value="Submit">
</form>
```

THE <FORM> ELEMENT

- The HTML **<form>** element defines a form that is used to collect user input
- An HTML form contains **form elements**.
- Form elements are different types of input elements, like text fields, checkboxes, radio buttons, submit buttons, and more.

```
<form>
.
form elements
.
</form>
```

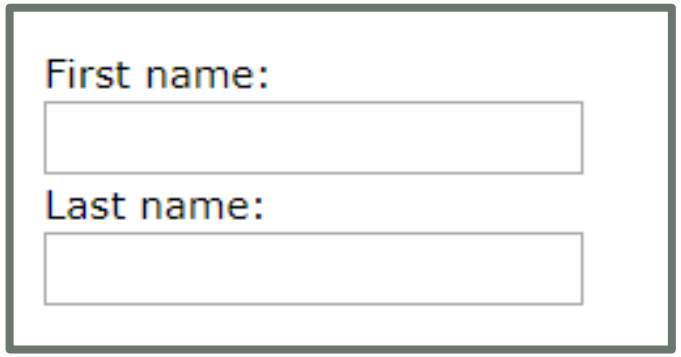
THE <INPUT> ELEMENT

- The **<input>** element is the most important form element.
- The <input> element can be displayed in several ways, depending on the **type** attribute.

Type	Description
<input type="text">	Defines a one-line text input field
<input type="radio">	Defines a radio button (for selecting one of many choices)
<input type="submit">	Defines a submit button (for submitting the form)

TEXT INPUT

<input type="text"> defines a one-line input field for **text input**



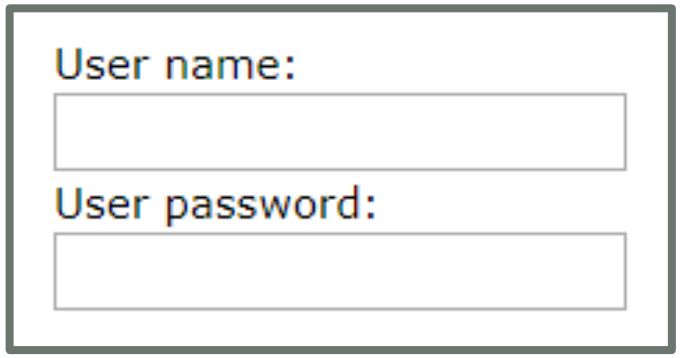

```
<form>
  First name:<br>
  <input type="text" name="firstname" value="Mickey"><br>
  Last name:<br>
  <input type="text" name="lastname" value="Mouse">
</form>
```

Note: The form itself is not visible. Also note that the default width of a text field is 20 characters.

PASSWORD INPUT

<input type="password"> defines a **password field**

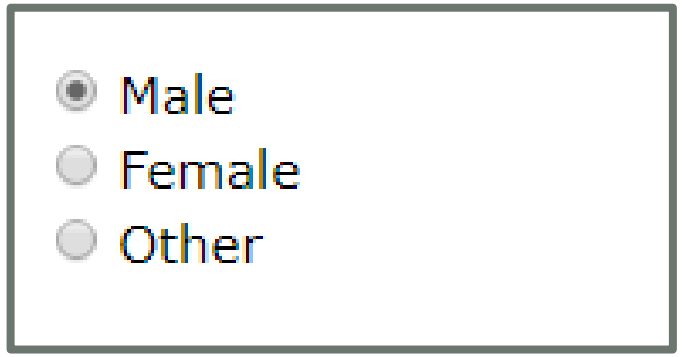
The characters in a password field are masked (shown as asterisks or circles).




```
<form>
        User name:<br>
        <input type="text" name="username"><br>
        User password:<br>
        <input type="password" name="psw">
</form>
```

RADIO BUTTON INPUT

<input type="radio"> defines a **radio button**.



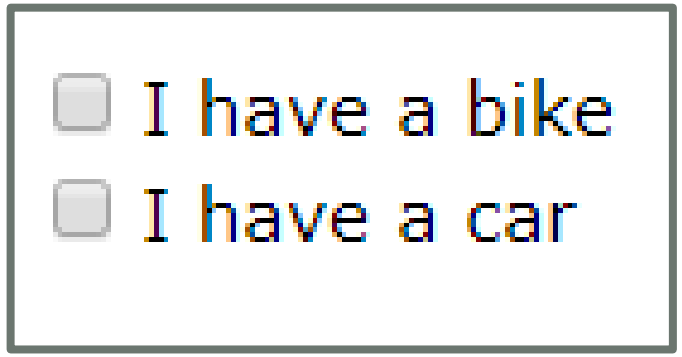

```
<form>
        <input type="radio" name="gender" value="male" checked> Male<br>
        <input type="radio" name="gender" value="female"> Female<br>
        <input type="radio" name="gender" value="other"> Other
</form>
```

Note: The form itself is not visible. Also note that the default width of a text field is 20 characters.

CHECKBOX INPUT

<input type="checkbox"> defines a **checkbox**.

Checkboxes let a user select ZERO or MORE options of a limited number of choices.




```
<form>
  <input type="checkbox" name="vehicle1" value="Bike"> I have a bike<br>
  <input type="checkbox" name="vehicle2" value="Car"> I have a car
</form>
```

THE VALUE ATTRIBUTE

The **value** attribute specifies the initial value for an input field

```
<form action="">
       First name:<br>
       <input type="text" name="firstname" value="John">
</form>
```

THE SUBMIT BUTTON

<input type="submit"> defines a button for **submitting** the form data to a **form-handler**.

- The form-handler is typically a server page with a script for processing input data.
- The form-handler is specified in the form's **action** attribute


```
<form action="action_page.php">
  First name:<br><input type="text" name="firstname" value="Mickey"><br>
  Last name:<br><input type="text" name="lastname" value="Mouse"><br><br>
  <input type="submit" value="Submit">
</form>
```

THE <SELECT> ELEMENT

The **<select>** element defines a **drop-down list**:

```
<select name="cars">
  <option value="volvo">Volvo</option>
  <option value="saab">Saab</option>
  <option value="fiat">Fiat</option>
  <option value="audi">Audi</option>
</select>
```

- The **<option>** elements defines an option that can be selected.
- By default, the first item in the drop-down list is selected.
- To define a pre-selected option, add the **selected** attribute to the option:

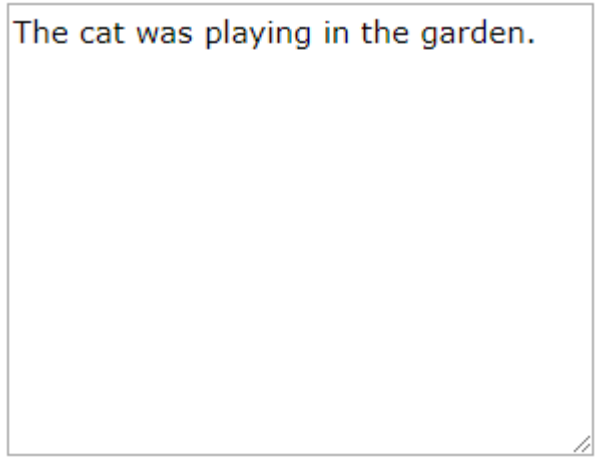
```
<option value="fiat" selected>Fiat</option>
```

THE <TEXTAREA> ELEMENT

The **<textarea>** element defines a multi-line input field (**a text area**)

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

- The **rows** attribute specifies the visible number of lines in a text area.
- The **cols** attribute specifies the visible width of a text area.
- This is how the HTML code above will be displayed in a browser



HTML BASIC TAGS

HTML button

- HTML button defines a clickable button.

```
<button type="button">Click Me!</button>
```

- Inside a <button> element you can put content, like text or images. This is the difference between this element and buttons created with the <input> element.
- **Tip:** Always specify the type attribute for a <button> element. Different browsers use different default types for the <button> element.

THE ACTION ATTRIBUTE

- The **action** attribute defines the action to be performed when the form is submitted.
- Normally, the form data is sent to a web page on the server when the user clicks on the submit button.
- In the example above, the form data is sent to a page on the server called "action_page.php". This page contains a server-side script that handles the form data
- If the action attribute is omitted, the action is set to the current page.

```
<form action="action_page.php">
```

THE METHOD ATTRIBUTE

The **method** attribute specifies the HTTP method (**GET** or **POST**) to be used when submitting the form data

```
<form action="action_page.php" method="get">
<form action="action_page.php" method="post">
```

GET VS. POST

When to Use GET?

- The default method when submitting form data is GET.
- GET must NOT be used when sending sensitive information! GET is best suited for short, non-sensitive, amounts of data, because it has size limitations too
- However, when GET is used, the submitted form data will be **visible in the page address field**

```
action_page.php?firstname=Mickey&lastname=Mouse
```

When to Use POST?

- Always use POST if the form data contains sensitive or personal information. The POST method does not display the submitted form data in the page address field.
- POST has no size limitations, and can be used to send large amounts of data.

THE NAME ATTRIBUTE

- Each input field must have a **name** attribute to be submitted.
- If the name attribute is omitted, the data of that input field will not be sent at all.
- This example will only submit the "Last name" input field

```
<form action="action_page.php">
       First name:<br>
       <input type="text" value="Mickey"><br>
       Last name:<br>
       <input type="text" name="lastname" value="Mouse"><br><br>
       <input type="submit" value="Submit">
</form>
```

THE END

WEEK3

PHP CONDITIONAL STATEMENTS

PHP CONDITIONAL STATEMENTS

In PHP we have the following conditional statements:

- 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

PHP - THE IF STATEMENT

The if statement executes some code if one condition is true.

Syntax

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

```
<?php
$n = 5;
if ($n < 10)
{
  echo " Number is less than ten ";
}
?>
```

```
Number is less than ten
```

$X==$Y true/False
$X=$Y

```php
<?php
$n = 5;
if ($n < 10)
{
  echo " Number is less than ten <br/>";
}
  echo “Hello”;
?>
```

```
Number is less than ten
Hello
```

```php
<?php
$n = 5;
if ($n < 10)
  echo " Number is less than ten <br/>";
  echo “Hello”;
?>
```

```
Number is less than ten
Hello
```

```php
<?php
$n = 5;
if ($n > 10)
  echo " Number is less than ten <br/>";
  echo “Hello”;
?>
```

```
Hello
```

```php
<?php
$n=30;
if($n>=50);
echo "Pass <br/>";
echo "class";
?>
```

```
Pass
Class
```

```php
<?php
$n=60;
if($n>=50);
echo "Pass <br/>";
echo "class";
?>
```

```
Pass
Class
```

```php
<?php
$n=60;
$k=0;
if($n>=50)
{
echo "Pass <br/>";
$k=5;
}
echo $k;
?>
```

```
Pass
5
```

```php
<?php
$n=30;
$k=0;
if($n>=50)
{
echo "Pass <br/>";
$k=5;
}
echo $k;
?>
```

```
0
```

```php
<?php
$n=60;
$k=0;

if($n>=50);
{
echo "Pass <br/>";
$k=5;
}
echo $k;
?>
```

```php
<?php
$n=30;
$k=0;

if($n>=50);
{
echo "Pass <br/>";
$k=5;
}
echo $k;
?>
```

```
Pass
5
```

```php
<?php
$n=60;
$k=0;
if($n<=50)
echo "Pass <br/>";
$k=5;
echo $k;
?>
```

```
5
```

PHP - THE IF...ELSE STATEMENT

The **if...else** statement executes some code if a condition is true and another code if that condition is false.

Syntax

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

```
<?php
$n = 5;
if ($n < 10)
{
  echo " Number is less than ten ";
}
else
{   echo " number is greater than or equal ten" ;
}
?>
```

```
Number is less than ten
```

```php
<?php
$n=5;
if($n>10)
echo "Number is less than ten <br/>";
else
echo “Number is greater than or equal ten";
?>
```

Number is less than ten

```php
<?php
$n=5;
if($n<10)
echo "Number is less than ten <br/>";
else
echo “Number is greater than or equal ten";
?>
```

Number is greater than or equal ten

```php
$n=5;
if($n>10);
echo "Number is less than ten <br/>";
else
echo "number is greater than or equal ten";
```

Error

```php
<?php
$n=5;
if($n>10)
echo "Number is less than ten <br/>";
else;
echo "Number is greater than or equal ten";
?>
```

```
Number is greater than or equal ten
```

```php
<?php
$n=5;
if($n<10)
echo "Number is less than ten <br/>";
else;
echo "Number is greater than or equal ten";
?>
```

```
Number is less than ten
Number is greater than or equal ten
```

```
$x="g";
if($x=="m" || $x=="M")
    echo "Male <br/>";
else
    echo "Female";
```

```
Female
```

PHP - THE IF...ELSEIF...ELSE STATEMENT

The **if...else** statement executes some code if a condition is true and another code if that condition is false.

Syntax

```
if (condition)
{
  code to be executed if this condition is true;
}
elseif (condition)
{
  code to be executed if first condition is false and this condition is true;
}
else
{
  code to be executed if all conditions are false;
}
```

PHP - THE IF...ELSEIF...ELSE STATEMENT

```
<?php
$n = 5;
if ($n < 10)
  {
  echo " Number is less than ten ";
  }
else if ( $n > 10 )
  {
  echo “number is greater than ten" ;
  }
 else
  {
  echo "number is aqual to ten" ;
  }
?>
```

EXAMPLES

```php
<?php
$a = 100;
$b = 50;

if ($a > $b) {
    echo "a is bigger than b";
} elseif ($a == $b) {
    echo "a is equal to b";
} else {
    echo "a is smaller than b";
}
?>
```

```
a is bigger than b
```

EXAMPLES

```
<?php
$a = 0;

if(++$a == 3) echo 3;
elseif(++$a == 2) echo 2;
elseif(++$a == 1) echo 1;
else echo "No match!";
?>
```

```
2
```

EXAMPLES

90-100	A
80-89	B
70-79	C
60-69	D
0-59	F

```
<?php
$M=70;
if($M>=90 && $M<=100)
echo "A <br/>";
else if($M>=80 && $M<=89)
echo "B <br/>";
else if($M>=70 && $M<=79)
echo "C <br/>";
else if($M>=60 && $M<=69)
echo "D <br/>";
else if($M>=0 && $M<=59)
echo "F <br/>";
else
echo "Invalid Mark <br/>";
?>
```

C

اكتب برنامج لحساب قيمة f(x) التالية اذا قيمة x=1:

EXAMPLES

$$f(x) = \begin{cases} 1, & x < 0 \\ 2x^2 + 5, & 0 \le x \le 2 \\ 3x, & x > 2 \end{cases}$$

```
<?php
$x=1;

if($x<0)
echo 1 . "<br/>";

else if($x>=0 && $x<=2)
echo 2*$x*$x+5 . "<br/>";

else if($x>2)
echo 3*$x. "<br/>";

?>
```

7

THE END