JAVASCRIPT ACADEMY

THE STRESS FREE-WAY TO LEARNING JAVASCRIPT INSIDE AND OUT

IT ACADEMY

JavaScript Academy

The Stress Free Way To Learning JavaScript Inside & Out

This document is geared towards providing exact and reliable information in regards to the topic and issue covered. The publication is sold with the idea that the publisher is not required to render accounting, officially permitted, or otherwise, qualified services. If advice is necessary, legal or professional, a practiced individual in the profession should be ordered.

- From a Declaration of Principles which was accepted and approved equally by a Committee of the American Bar Association and a Committee of Publishers and Associations.

The information provided herein is stated to be truthful and consistent, in that any liability, in terms of inattention or otherwise, by any usage or abuse of any policies, processes, or directions contained within is the solitary and utter responsibility of the recipient reader. Under no circumstances will any legal responsibility or blame be held against the publisher for any reparation, damages, or monetary loss due to the information herein, either directly or indirectly.

Table of Contents

Introduction

I want to thank you and congratulate you for downloading the book, *School of JavaScript.*

This book contains proven steps and strategies on how to master the basics of the JavaScript language.

This eBook is created for people who want to learn JavaScript quickly and easily. By focusing on the most important aspects of JavaScript, this book will teach you what you need to know in just a few pages. This is the only book you ' need if you want to understand the basics of the JavaScript programming language.

Thanks again for downloading this book, I hope you enjoy it!

Bonus: As a thank you I'd like to offer you a gift. I've included a few bonus books. Check out the table of contents or go to the very botom to find out how to get these!

Chapter 1: The JavaScript Language

Basic Information

JavaScript is one of the most popular programming languages today. This language plays an important role in the development of modern websites and webpages. Its characteristics and capabilities can help you create dynamic websites that support user interactions. Additionally, JavaScript is an interpreted computer language that possesses powerful object-oriented features.

This language was originally called “ LiveScript, ” but NetScape (the company that created it) changed the name to JavaScript in order to exploit the popularity enjoyed by Java (another programming language). It must be pointed out that JavaScript has no connection with the Java language.

The Client-Side Form of JavaScript

According to computer experts, the most popular variant of this language is the Client-Side JavaScript. Website developers use this variant in creating their projects. In general, you may use Client-Side JavaScript in two different ways:

- Include it in the codes of the webpage/s
- Save it in an external file

For the second option, you will need to use an HTML document to reference the external file and link it to your website.

Basically, you don ’ t have to use static HTML in creating your webpages. You can include apps and programs that allow browser control, user interactions, and dynamic creation of HTML files.

The Client-Side type of JavaScript has various advantages over typical CGI (i.e. Common Gateway Interface) server-side scripts. For instance, you can use Client-Side JavaScript to determine if the user has provided valid contact information in your form fields.

JavaScript codes run whenever a user submits a form. Then, the system will analyze the information in the submitted form. This mechanism ensures that the Web Server won ' t process invalid requests. Thus, Client-Side JavaScript offers better security and performance for your websites.

This programming language also allows you to collect implicit or explicit actions that users initiate (e.g. link navigation, button clicks, etc.).

The Advantages Offered by JavaScript

Here are the main advantages offered by this programming language:

- You can reduce server interactions – JavaScript can validate the information entered by the users before sending it to the web server. That means you can filter out the useless traffic from your server.
- Instant feedback – Your website visitors will immediately see if they have missed a field or entered the wrong data. They will see the error message without any page reload.
- Improved webpage interactivity – With JavaScript, you can generate user interfaces that respond to mouse and keyboard inputs.
- Enhanced Interfaces – In this language, you can easily add cool interface items (e.g. sliders, drag-and-drop components, etc.) to provide your website visitors with an enhanced interface.

The Disadvantages of JavaScript

Just like other things in life, JavaScript has its own drawbacks. The main ones are:

- This language doesn ’ t have multiprocessing or multithreading capabilities.
- You can ’ t use JavaScript for networking programs. This language doesn ’ t support programs/applications that focus on networking.
- You can ’ t read or edit files written using Client-Side JavaScript. These features are disabled for security reasons.

Chapter 2: The Syntax of JavaScript

You may implement this language by placing JavaScript statements inside your webpage ’ s **<script>** tags. In general, you can put these **<script>**tags anywhere in the webpage. Professional website developers, however, usually keep these tags inside the page ’ s **<head>** tags.

Basically, a <script> tag instructs browsers to interpret the text it contains as a script. Here is the basic syntax of JavaScript:

```
<script ...>
  JavaScript code
</script>
```

Script tags accept the following attributes:

- Language – This attribute indicates the language being used. Recent versions of HTML and XHTML, however, no longer support this attribute.
- Type – This is the attribute being used in modern sites and webpages. You must use it to indicate the computer language you are using. Make sure that its value is “ text/javascript ” .

Taking those attributes into account, the syntax of JavaScript is:

```
<script language="javascript" type="text/javascript">
  JavaScript code
</script>
```

Using JavaScript for the First Time

In this section, you ’ ll be using this scripting language for the first time. The code that you will use is designed to display “ Hello World ” on the screen of your computer. As you can see, the JavaScript code (see the screenshot below) involves an optional HTML element. That HTML comment allows you to save the code from browsers that don ’ t support the JavaScript language.

The symbol “ //→ ” terminates the HTML comment. In JavaScript, “ // ” represents comments. That means you have to use a pair of slashes to prevent browsers from interpreting HTML comments as part of your JavaScript statements. Then, you must invoke the **document.write** function to write a string into your HTML file.

Important Note: You can use the document.write function to write HTML statements, text-based content, or both.

Here ’ s the code that you must type into your JavaScript editor:

```
<html>
<body>
<script language="javascript" type="text/javascript">
<!--
   document.write ("Hello World!")
//-->
</script>
</body>
</html>
```

If written and executed properly, the code given above will show you: **Hello World!**

Whitespaces and Line Breaks

This scripting language ignores tabs, spaces, and newlines inside your programs. That means you may use these elements to make your JavaScript codes readable and understandable.

Semicolons

Similar to other computer languages (e.g. C++), JavaScript requires you to terminate simple statements using a semicolon. However, you won ’ t have to add a semicolon if your statements are entered on separate lines. For instance, the code below works fine even if it doesn ’ t have any semicolon:

```
<script language="javascript" type="text/javascript">
<!--
  var1 = 10
  var2 = 20
//-->
</script>
```

The Case-Sensitivity of JavaScript Statements

This programming language is case-sensitive. For example, “ WORD, ” “ Word, ” and “ word ” are treated as three different entities in JavaScript. That means you must be consistent with letter capitalization when working on functions, keywords, variables, and other identifiers.

JavaScript Comments

You must remember the following rules when creating a comment in JavaScript:

- This language ignores all of the text placed between the “ // ” symbol and the end of a line.
- You may use /* and */ to create comments. In general, programmers employ these symbols when creating multi-line comments.
- JavaScript also allows you to use <! — (i.e. the symbol used in HTML to start comments) when writing single-line comments.
- JavaScript doesn ’ t recognize → (i.e. the symbol used in HTML to end comments). You must use //→ instead.

Chapter 3: The Placement of JavaScript Codes

You can place JavaScript codes in any part of an HTML file. However, the recommended areas for keeping JavaScript codes inside an HTML document are:

- In the <head> </head> area
- In the <body> </body> area
- In the <head> </head> and <body> </body> areas of the document
- Inside an external document

This chapter will discuss each of these areas.

Placing Your Code in the <head> </head> Area

This is the ideal area if you want your scripts to run during specific events (e.g. when the user clicks on a button). The sample code below will show you how to save codes in this area.

```
<html>
<head>
<script type="text/javascript">
<!--
function sayHello() {
   alert("Hello World")
}
//-->
</script>
</head>
<body>
Click here for the result
<input type="button" onclick="sayHello()" value="Say Hello" />
</body>
</html>
```

If entered correctly, the code will show you this:

Placing Your Code in the <body> </body> Area

In general, you should place a JavaScript code in the <body> section if you want it to produce content as the webpage loads. Here, you don ’ t have to define any function using JavaScript. Here ’ s an example:

```
<html>
<head>
</head>
<body>
<script type="text/javascript">
<!--
document.write("Hello World")
//-->
</script>
<p>This is web page body </p>
</body>
</html>
```

That code generates the following result:

```
Hello World
This is web page body
```

Placing Your Code in the <head> </head> and <body> </body> Areas of the Document

You also have the option to place your JavaScript code in both the <head> and <body> areas of the HTML file. Analyze the following example:

```
<html>
<head>
<script type="text/javascript">
<!--
function sayHello() {
   alert("Hello World")
}
//-->
</script>
</head>
<body>
<script type="text/javascript">
<!--
document.write("Hello World")
//-->
</script>
<input type="button" onclick="sayHello()" value="Say Hello" />
</body>
</html>
```

If you entered the code properly, opening the HTML file will give you this:

Placing the Code in an External File

While using JavaScript, you ' ll surely find many situations where you have to use the same codes on different webpages. Typing the same code repeatedly can be boring and time-consuming. Fortunately, you can simply save your JavaScript codes in an external file. Then, you just have to link that file to the webpages that need the identical codes.

In using this method, you should use a **script** tag and the **src** attribute. Check the example below:

```
<html>
<head>
<script type="text/javascript" src="filename.js" ></script>
</head>
<body>
.......
</body>
</html>
```

This method requires you to write your JavaScript code inside a text file and save it using " .js " as the extension (e.g. example.js).

Let ' s try a simple exercise. Copy the code below into a text file and save it as " filename.js " . Then, utilize the **sayHello** function of your HTML document after adding the filename.js file.

```
function sayHello() {
   alert("Hello World")
}
```

Chapter 4: The Variables in the JavaScript Language

The Different Data Types in JavaScript

Similar to other programming languages, JavaScript has its own set of supported data types. Thus, there are certain values and information that you can represent and manipulate using JavaScript.

This programming language supports these basic data types:

- Boolean (i.e. true or false)
- Strings (e.g. “ This book is awesome. ”)
- Numbers (e.g. 1, 2, 3, 123, etc.)

JavaScript also supports **null** and **undefined,** two data types that describe a single value. Lastly, this language uses a complex type of data called “ object. ” You ’ ll learn about objects later in this book.

Important Note: JavaScript treats all numbers as floating-point values.

The Variables

As a JavaScript user, you may think of variables as containers: you may place different types of information in a variable. To access the stored data, you just have to specify the container that holds it.

You have to declare variables before you can use them. In JavaScript, you should use **var** when declaring a variable. Check the following example:

```
<script type="text/javascript">
<!--
var money;
var name;
//-->
</script>
```

The code given above declares two variables: **money** and **name**.

The process of saving data into a variable is known as variable initialization. Most programmers initialize variables during the declaration. However, you may simply declare a variable without saving any value: you can save data into the variable once you need to.

For example, you can create a variable called **profit**without entering any value. Once you have deducted your expenses from your gross income, you may go back to the variable you created and enter the information. To initialize a variable, you just have to indicate the variable ’ s name, place an equal sign, and enter the value you want to assign. Here ’ s an example:

var profit;

profit = 1,000,000.00;

JavaScript is considered as an “ untyped ” computer language. That means JavaScript variables can contain any type of data. This is a great advantage over other languages,

which require users to specify the data type during variable creation.

The Scope of a Variable

A variable ' s " scope " is the area of your computer program in which it is declared. JavaScript supports the following scopes:

- Local – Local variables are visible only inside the function where they were defined. That means a function won ' t be able to use the local variable of another function.
- Global – Global variables are available for all the functions inside the program. That means you can define them anywhere in your codes.

If a local variable and a global variable share the same name, the local variable will take precedence. If you ' ll declare a variable locally using the name of an existing global variable, you ' ll be hiding that global variable. Analyze the sample code below:

```
<script type="text/javascript">
<!--
var myVar = "global"; // Declare a global variable
function checkscope( ) {
   var myVar = "local";  // Declare a local variable
   document.write(myVar);
}
//-->
</script>
```

The code given above will show you this result: **Local**.

Naming Your Variables

While choosing a name for your variables, you should remember the following rules:

- You can ’ t use any JavaScript keyword (e.g. break, Boolean, etc.) as the name of a variable. You ’ ll learn about JavaScript keywords later.
- You can use letters, numbers, and an underscore when naming a variable. However, you can ’ t begin the name using a number. For example, **sample1** is valid but **1sample**isn ’ t.
- The names of variables are case-sensitive. That means JavaScript treats “ SAMPLE, ” “ Sample, ” and “ sample ” as three different variables.

The Reserved Keywords in JavaScript

Similar to other programming languages, JavaScript has a collection of words that serve special purposes. You can ’ t use these words in naming objects, methods, functions, loops, and variables. The table below will show you all of the keywords in the JavaScript language.

if	do	in	int
new	try	for	var
else	enum	this	long
byte	case	true	null
char	goto	void	with
break	throw	false	final

catch	float	class	const
while	short	super	switch
export	throws	typeof	public
return	delete	static	double
import	Instanceof	abstract	synchronized
Boolean	interface	extends	transient
finally	package	private	protected
function	volatile	continue	debugger
implements			

Chapter 5: The JavaScript Operators

The JavaScript language supports the following operator types:

- Logical Operators
- Comparison Operators
- Assignment Operators
- Arithmetic Operators
- Bitwise Operators
- Miscellaneous Operators

Let ’ s discuss each type of operator:

The Logical Operators

JavaScript supports these logical operators:

Note: Let ’ s assume that x = 5 and that y = 10.

- “ && ” – This is known as the **Logical AND** operator. If both operands are not equal to zero, the condition is true. For example: (x && y) is true.
- “ || ” – This operator is called **Logical OR**. If at least one of the operands is not equal to zero, the condition is true. For example: (x || y) is true.
- “ ! ” – Programmers refer to this operator as Logical NOT. This operator reverses the result of the operator it is used on. For instance: ! (x && y) is false.

The Comparison Operators

Here are the comparison operators that you can use in JavaScript:

Note: Let ’ s use the following variables: x = 2; y = 4.

- “ == ” – This operator checks the equality of the two operands. If the values are equal, the condition is true. For example: (x == y) is false.
- “ != ” – With this operator, you can check if the operands have unequal values. If the values are unequal, the condition is true. For instance: (x != y) is true.
- “ > ” – Here, you ’ ll check if the left-hand operand ’ s value is greater than that of the right-hand one. If it is, the condition is true. For instance, (y > x) is true.
- “ < ” – This operator checks the values of both operands. If the value of the left-hand operand is less than that of the right-hand operand, the condition is true. For example: (x > y) is true.
- “ >= ” - You can use this operator to check if the left-hand operand ’ s value is greater than or equal to that of the right-hand one. If it is, the condition is true. For instance: (y >= x) is true.
- “ <= ” – With this operator, you can check if the value of the left-hand operand is less than or equal to that of the right-hand operand. If it is, the condition is true. For example: (x <= y) is true.

The Assignment Operators

These are the operators that you can use to assign values:

- " = " - This is the **Simple Assignment** operator. It assigns the value of the right-hand operand to the left-hand operand. For example: (z = x + y) assigns the value of x + y to z.
- " += " – This operator is known as " **Addition and Assignment** " operator. It adds the values of both operands and gives the result to the left-hand operand. For example: (z += x) is equal to (z = z + x).
- " -= " – Programmers call this the " **Subtraction and Assignment** " operator. It subtracts the value of the right-hand operand from that of the left-hand operand. Then, it assigns the difference to the left-hand operand. For example: (z -= x) is equal to (z = z – x).
- " *= " – This operator is called " **Multiplication and Assignment**. " It multiplies the values of both operands and gives the product to the left-hand operand. For example: (z *= x) is equal to (z = z * x).
- " /= " – This is known as the " **Division and Assignment** " operator. It divides the value of the left-hand operand by the value of the right-hand operand. Then, it assigns the quotient to the left-hand operand. For instance: (z /= x) is equal to (z = z / x).
- " %= " – JavaScript users refer to it as the " **Modulus and Assignment** " operator. It takes the modulus of both operands. Then, it gives the result to the left-hand operand. For example: (z %= x) is equal to (z = z % x).

The Arithmetic Operators

The list below shows the arithmetic operators available in JavaScript. To help you understand how these operators work, let ’ s use two sample variables: x = 2; y = 4.

- “ + ” – You should use this operator if you want to perform addition on two operands (i.e. the values on either side of the operator). For instance: x + y = 6.
- “ - ” – This operator allows you to perform subtraction on your codes. Here, you ’ ll deduct the value of the second operand from that of the first. For example: y – x = 2.
- “ * ” – You should use an asterisk if you want to multiply the two operands. For example: x * y = 8.
- “ / ” – With this operator, you ’ ll divide the value of the left-hand operand from that of the right-hand operand. For example: y / x = 2.
- “ ++ ” – This operator allows you to add 1 to the operand it is used on. For example: 4++ = 5.
- “ -- ” – This operator subtracts 1 from the operand it is attached to. For example: x— = 1.

Important Note: You may also use the “ + ” operator to combine numeric and string elements. For example, “ x ” + 4 = x4.

The Bitwise Operators

This programming language allows you to perform bitwise operations in your codes. Here are the operators that you can use:

Note: Let ’ s use two variables: x = 2; y = 3.

- “ & ” – This is called the **Bitwise AND** operator. It conducts the **Boolean AND** operation on all of the bits involved in its arguments. For example: (x & y) is 2.
- “ | ” – Programmers refer to this operator as **Bitwise OR**. It conducts the **Boolean OR** operation on the bits involved in its arguments. For instance: (x | y) is 3.
- “ ^ ” – This is known as the **Bitwise XOR** operator. It conducts the **Boolean Exclusive OR** operation on all of the bits involved in its arguments. With **Exclusive OR**, only one of the operands can be true. For example: (x ^ y) is 1.
- “ ~ ” – This operator is “ unary ” (i.e. it works on a single operand). Known as the **Bitwise NOT** operator, it reverses the bits of the operand. For example (~y) is -4.
- “ << ” – This operator is called **Left Shift**. It moves the bits of the left-hand operand to the left based on the number specified by the right-hand operand. Also, it uses zeros to fill new bits. Moving a value to the left by 1 position is like multiplying it by 2. Thus, (x << 1) gives 4.
- “ >> ” - This is known as the **Right Shift** operator. It moves the value of the left-hand operand to the right based on the number given by the right-hand operand. For example: (x >> 1) gives 1.
- “ >>> ” – JavaScript users call this “ **Right Shift and Zero**. ” Its function is

similar to that of “ >>. ” The only difference is that the leftmost bits are always equal to zero.

The Miscellaneous Operators

This section of the book will discuss two useful operators that don ' t belong to the categories discussed above. These operators are:

- " ?: " – This is known as the " **Conditional Operator**. " It checks whether the value of an expression is true or false. Then, it performs one of the assigned statements based on the evaluation ' s result.

  For example, if the condition is true, the operator will assign 1 to the expression. If the condition is false, however, the operator will assign 2.

- typeof – This is a unary operator, which means you can use it on a single operand. You can use it to determine an operand ' s data type. This operator gives " string, " " boolean, " or " number " if the operand is a string, Boolean, or number. Then, it gives true or false, depending on the result of the evaluation.

The table below provides useful information regarding the typeof operator:

Data Type	The typeof Result
Object	" object "
Number	" number "
Boolean	" boolean "
Undefined	" undefined "
Null	" object "

Function	“ function ”
String	“ string ”

Chapter 6: The Conditional Statements

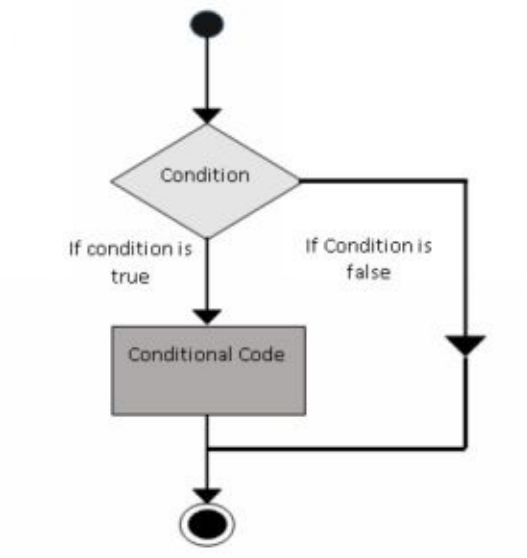
While using JavaScript, there will be situations that require you to choose one out of a predetermined set of choices. In these cases, you must use conditional statements. A conditional statement allows your program to make proper judgments and execute correct processes.

This programming language offers conditional statements that you can use on your codes. This chapter will discuss three of the most important conditional statements: (1) the “ if ” statement, (2) the “ if … else ” statement, and (3) the “ if … else if …” statement.

Let ’ s discuss these statements in detail:

The Flow Chart of a Conditional Statement

The image below shows the basic flow chart of a conditional statement.



The if Statement

This statement serves as the basic conditional statement in JavaScript. It helps programs in making decisions and executing other statements.

The syntax of the “ if ” statement is:

```
if (expression){
   Statement(s) to be executed if expression is true
}
```

The code given above evaluates JavaScript expressions. If the result of the evaluation is true, the assigned statement/s will be performed. If the result is false, however, none of the assigned statements will be performed. In most cases, you ’ ll be using comparison operators with this statement.

The if … else Statement

This statement is more complex than the previous one. That ’ s because the “ if … else ” statement offers better control in executing statements conditionally.

Here is the syntax of the “ if … else ” statement:

```
if (expression){
   Statement(s) to be executed if expression is true
}else{
   Statement(s) to be executed if expression is false
}
```

You can use the syntax given above to evaluate JavaScript expressions. If the result is true, the statement placed in the “ if ” section will be performed. If the result is false, on the other hand, the statement within the “ else ” section will be performed.

The “ if … else if …” Statement

Basically, this is an enhanced form of the “ if … else ” statement. By providing multiple conditions, it helps your programs in making the right decisions and performing the correct actions.

Here is the syntax that you must use when creating an “ if … else if …” statement:

```
if (expression 1){
   Statement(s) to be executed if expression 1 is true
}else if (expression 2){
   Statement(s) to be executed if expression 2 is true
}else if (expression 3){
   Statement(s) to be executed if expression 3 is true
}else{
   Statement(s) to be executed if no expression is true
}
```

If you ’ ll think about it, the syntax given above is a basic one. It ’ s a simple collection

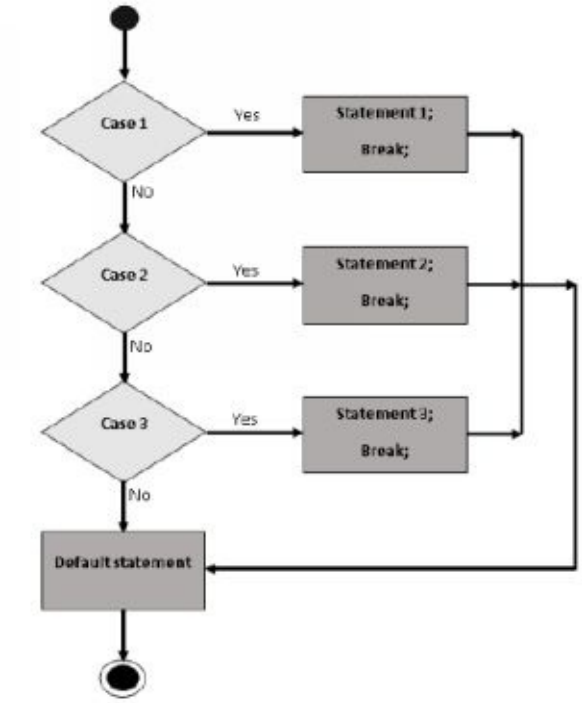
of “ if ” statements where every “ if ” serves as a part of the preceding else statement ’ s else clause. This syntax executes statement/s depending on the result of the evaluation. If all of the given conditions are false, the system will run the “ else ” section of the code**.**

The Switch-Case Statements

You may use a series " if … else if " statements to execute a multi-path branch system. You ' ve learned about this in the previous section of the book. However, this method isn ' t always the ideal answer, particularly if all of the branches rely on the value of one variable.

Fortunately, JavaScript offers switch statements. A single switch statement is more effective than a group of " if … else if " statements when used in the situation described above.

Here is the flowchart of a basic switch-case statement:



The Syntax

Switch statements are designed to do two things:

- Provide an expression that the program can evaluate
- Provide various statements that will run depending on the result of the evaluation.

JavaScript ’ s built-in interpreter compares each case with the expression ’ s value. This process will continue until the interpreter finds a match. If no match is found, the interpreter will run the default condition. The syntax of a switch statement is:

```
switch (expression)
{
  case condition 1: statement(s)
                    break;
  case condition 2: statement(s)
                    break;
   ...
  case condition n: statement(s)
                    break;
  default: statement(s)
}
```

This syntax uses break statements to terminate each case. If you ’ ll remove these break statements, the JavaScript interpreter will execute all of the statements inside each case. You ’ ll learn about break statements in the next chapter.

Chapter 6: The Different Types of Loops in JavaScript

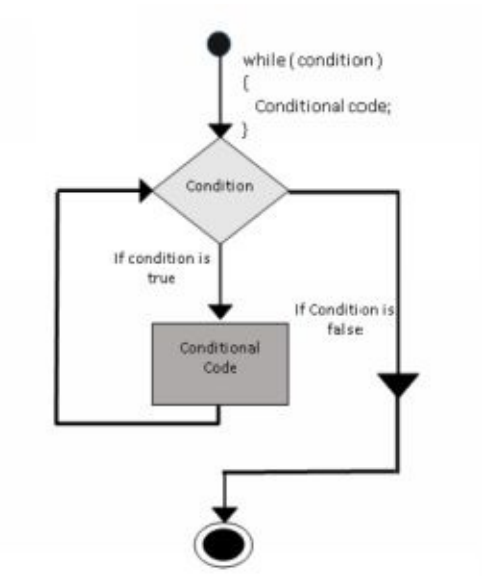
When writing your own programs, you ’ ll encounter situations where you have to execute an action multiple times. Obviously, typing the same thing repeatedly can be time-consuming (and extremely boring). Fortunately, JavaScript offers various tools that can help you repeat statements quickly and easily. These tools, known as loops, are the main topic of this chapter.

Here are the loops that you can use in the JavaScript language:

While Loops

Programmers consider this as the most basic loop type in JavaScript. Basically, while loops execute a statement or block of code while an expression is true. The loops will end as soon as the expression becomes false.

Here is the flowchart of a while loop:



The Syntax

Here is the syntax of a while loop:

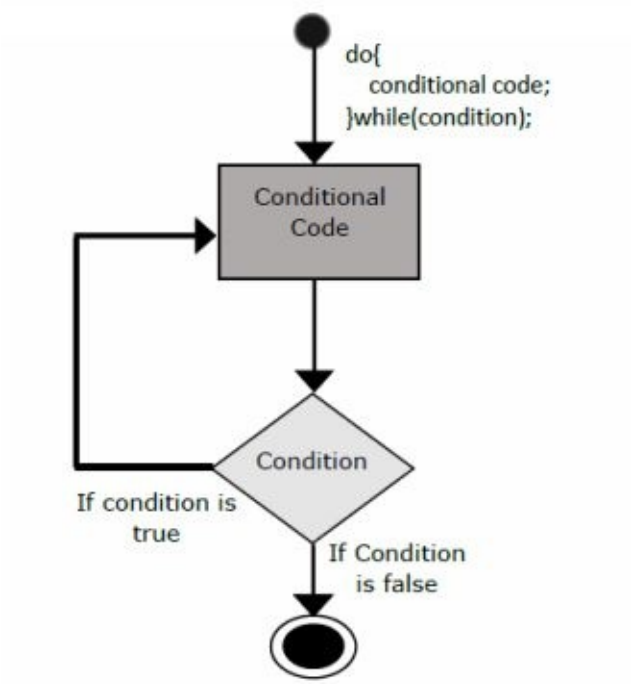
```
while (expression){
   Statement(s) to be executed if expression is true
}
```

Before executing any statement, a while loop evaluates the condition of the assigned expression. That means if the expression is false, none of the statements will be performed.

Do … while Loops

A “ do … while ” loop is similar to a while loop. The only difference is that it evaluates the expression after executing the assigned statement/s. If the result of the evaluation is true, the loop will execute the statement/s again. The process will only stop once the result becomes false. Because of this, a “ do … while ” loop will run at least once even if the result of the evaluation is false.

The flowchart of a typical “ do … while ” loop is:



The Syntax

Here is the syntax of a “ do … while ” loop:

```
do{
   Statement(s) to be executed;
} while (expression);
```

Important: The semicolon at the end of the syntax is mandatory. Make sure that you ’ ll include it in all of your “ do … while ” loops.

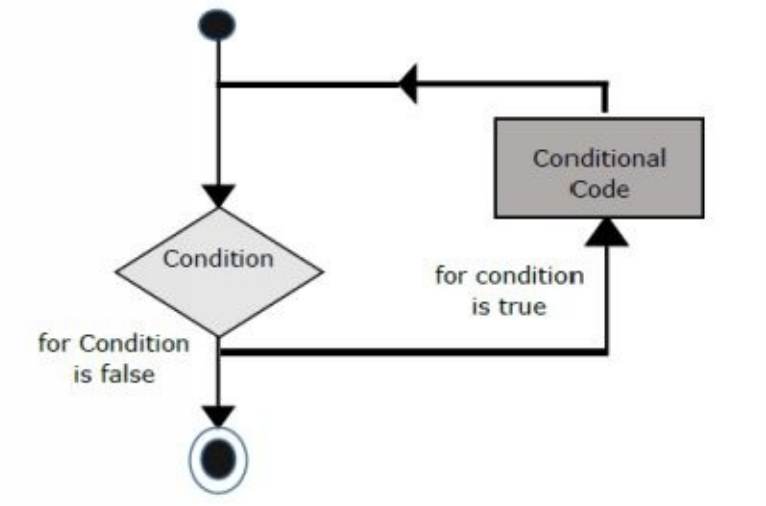
For Loops

This kind of loop is known for its compactness. In general, “ for loops ” have the following parts:

- Loop Initialization – This is the part where you begin your counter at an initial value. The interpreter of JavaScript executes this part before running the loop itself.
- Test Statement – This part of the loop evaluates the assigned condition. If the assigned condition is true, the interpreter will execute the code within the “ for ” loop. If the condition is false, however, the system control will be given to the statement after the “ for ” loop.
- Iteration Statement – You can use this part to increase or decrease the loop ’ s counter.

JavaScript allows you to place all of these parts in a single line. You just have to separate each part using semicolons.

The image below shows the flowchart of a “ for ” loop:



The Syntax

The syntax of a “ for ” loop is:

```
for (initialization; test condition; iteration statement){
     Statement(s) to be executed if test condition is true
}
```

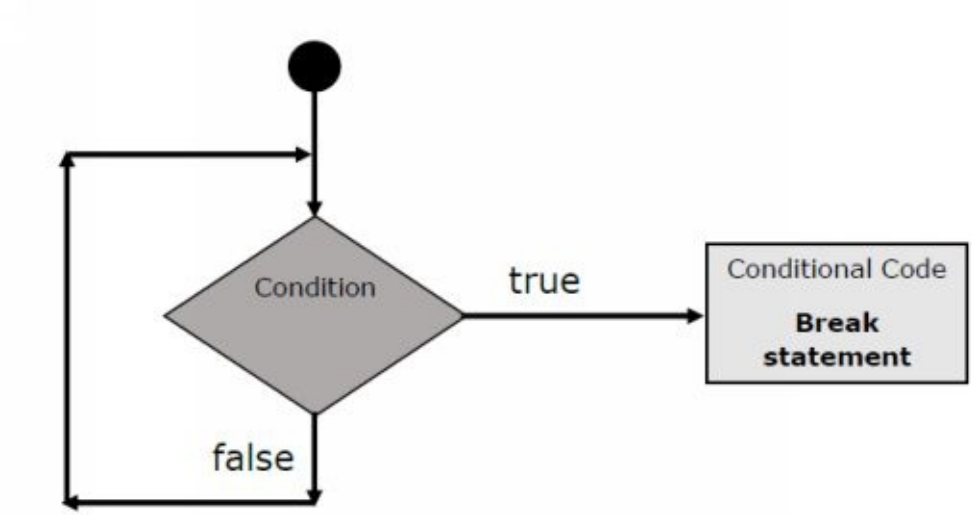
How to Control Your Loops

The JavaScript language allows you to manipulate all of your loops and switches. In some cases, you have to exit a loop statement before reaching its final section. There are also some situations where you have to ignore a certain part of the code and initialize the next one.

To help you with such situations, this programming language offers two powerful tools: the **break** statement and the **continue**statement. Let ' s discuss each statement in detail.

Break Statements

This statement, which was introduced in the previous chapter, allows you to exit any loop early. The image below will show you the flowchart of a basic break statement.



Note: You ' ve seen how break statements work during our discussion about switch statements.

Continue Statements

A **continue** statement instructs the JavaScript interpreter to begin the next iteration of a loop and ignore the remaining blocks of code. Once the system encounters a continue statement, the flow of the program goes to the expression evaluation. If the condition is still true, the continue statement will begin the next iteration. If the condition becomes

false, on the other hand, program control will go outside the loop.

Chapter 7: The Functions in the JavaScript Language

Basically, functions are collections of reusable codes that can be invoked at any part of your program. That means you won ’ t have to write the same code multiple times. Functions help programmers in creating modular codes. They also allow programmers to divide a large program into smaller functions.

Just like any other programming language, JavaScript offers features that can help in writing modular codes through functions. You ’ ve probably seen functions such as **write()** and **alert()** in the codes given in this book. You can use these functions repeatedly, but they were written in JavaScript once.

This language also allows you to write your own functions. This chapter will teach you the basics of functions in JavaScript.

How to Define a Function

You have to define functions before you can use them. The most popular method of defining a function involves the use of **function** (i.e. a JavaScript keyword). After typing **function**, specify the name and parameters that you ' d like to use. Then, add a block of JavaScript statement enclosed by braces.

Note: You may create a function without defining any parameter. Thus, the list of parameters may be empty.

Here is the syntax that you must use when defining a function:

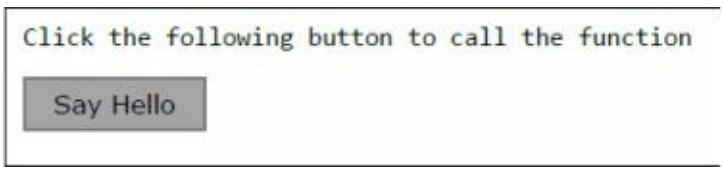
```
<script type="text/javascript">
<!--
function functionname(parameter-list)
{
  statements
}
//-->
</script>
```

How to Call a Function

To call a function in your scripts, you just have to specify the function ’ s name. You may use the code below as an example:

```
<html>
<head>
<script type="text/javascript">
function sayHello()
{
  document.write ("Hello there!");
}
</script>
</head>

<body>
<p>Click the following button to call the function</p>
<form>
<input type="button" onclick="sayHello()" value="Say Hello">
</form>

<p>Use different text in write method and then try...</p>
</body>
</html>
```

That code will give you this result:



The Parameters of a Function

The functions you ’ ve seen so far don ’ t have any parameter. However, you can easily assign various parameters while invoking/calling a function. You can capture these parameters within the function. Once captured, you may modify those parameters according to your needs.

Return Statements

JavaScript allows you to add a **return** statement to your functions. This **return** statement is completely optional, although you have to use it if you want to acquire values from your function/s. In general, you should place the return statement at the last section of your function.

For instance, you may pass two values into your function. Then, you may expect that function to multiply the values and show the result in the program you are calling.

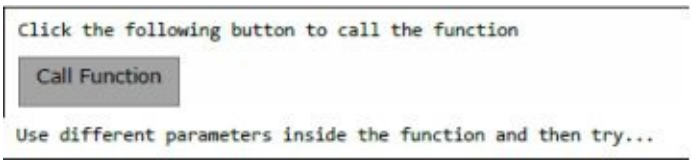
The Nested Functions

JavaScript allows you to place a function inside another function. This process is called “ nesting a function. ” As a general rule, you can ’ t include a function definition inside loops or conditional statements.

The code below will teach you how to nest your own functions:

```
<head>
<script type="text/javascript">
<!--
function hypotenuse(a, b) {
   function square(x) { return x*x; }

   return Math.sqrt(square(a) + square(b));
}
function secondFunction(){
   var result;
   result = hypotenuse(1,2);
   document.write ( result );
}
//-->
</script>
</head>

<body>
<p>Click the following button to call the function</p>

<form>
<input type="button" onclick="secondFunction()" value="Call Function">
</form>

<p>Use different parameters inside the function and then try...</p>
</body>
</html>
```

The code above gives the following result:



How to Create Functions using " Function() "

Aside from the function statement discussed above, you may also create functions by using **Function()** (i.e. a JavaScript constructor) and **new** (i.e. a JavaScript operator).

To use this constructor, you must use the following syntax:

```
<script type="text/javascript">
<!--
var variablename = new Function(Arg1, Arg2..., "Function Body");
//-->
</script>
```

This syntax can contain any amount of string arguments. The final argument serves as the function ' s body – it may hold arbitrary statements that are separated by semicolons.

Important Note: **Function()**doesn ' t specify names for the functions it generates. Programmers refer to these unnamed functions as " anonymous " functions.

Conclusion

Thank you again for downloading this book!

I hope this book was able to help you learn the fundamentals of JavaScript.

The next step is to write your own webpages or programs using the JavaScript programming language.

Finally, if you enjoyed this book, then I ’ d like to ask you for a favor, would you be kind enough to leave a review for this book on Amazon? It ’ d be greatly appreciated!

Click here to leave a review for this book on Amazon!

Thank you and good luck!

JavaScript Bootcamp

Learn the Basics of JavaScript Programming in 2 Weeks

This document is geared towards providing exact and reliable information in regards to the topic and issue covered. The publication is sold with the idea that the publisher is not required to render accounting, officially permitted, or otherwise, qualified services. If advice is necessary, legal or professional, a practiced individual in the profession should be ordered.

- From a Declaration of Principles which was accepted and approved equally by a Committee of the American Bar Association and a Committee of Publishers and Associations.

The information provided herein is stated to be truthful and consistent, in that any liability, in terms of inattention or otherwise, by any usage or abuse of any policies, processes, or directions contained within is the solitary and utter responsibility of the recipient reader. Under no circumstances will any legal responsibility or blame be held against the publisher for any reparation, damages, or monetary loss due to the information herein, either directly or indirectly.

The information herein is offered for informational purposes solely, and is universal as so. The presentation of the information is without contract or any type of guarantee assurance.

Table Of Contents

Introduction

I want to thank you and congratulate you for downloading the book, *"Learn the Basics of JavaScript Programming in 2 Weeks.*

This book contains proven steps and strategies on how to master the fundamentals of this powerful language.

This eBook will help you master the basics of JavaScript in two weeks. To make that possible, it focuses on the most important elements of this programming language. You won't read any irrelevant material while using this book. That means you will learn what you need to learn in order to become a proficient JavaScript user in just 14 days.

In this book you will learn:

What JavaScript is

How to use JavaScript

Common JavaScript data

And much more!

Thanks again for downloading this book, I hope you enjoy it!

Chapter 1: JavaScript and Codes

JavaScript – General Information

JavaScript was launched back in 1995 as a method of adding programs to webpages. Back then, it was exclusively used for a web browser called Netscape Navigator. Because of the success attained by JavaScript, all major browsers are now supporting it. This language allows programmers to create modern web programs – programs you can interact with without reloading webpages for each action. Traditional websites, on the other hand, also use JavaScript to provide different types of functionality and interactivity.

At this point, you should know that JavaScript is not related with Java (i.e. one of the leading programming languages today). JavaScript was called “Mocha” and “LiveScript” during its first year of existence. However, because the Java programming language was extremely popular, the owners of JavaScript decided to adopt its name. It was a marketing maneuver designed to exploit the popularity enjoyed by Java.

Codes – General Information

Codes are the texts that form computer programs and web applications. This eBook contains lots of codes, so you should be familiar with them. Computer experts claim that writing codes and reading codes are important aspects of becoming a programmer. Thus, you should read and understand the codes included in this book. You’ll be robbing yourself of excellent learning opportunities if you will simply glance over the code samples.

Beginners might experience difficulties in analyzing the codes. However, they will surely understand how JavaScript works if they will read the instructions carefully. It would be best if you will try the code samples using a JavaScript interpreter. This way, you can receive instant information about how codes look like once they are compiled as a

program.

Chapter 2: The Different Aspects of JavaScript

Data and Bits

Data forms the digital world. If you'll "visit" the world inside a computer, there is one thing you will see: data. You may read, edit, and generate data. Whenever you save new data, it will be stored as sequences of bits. This is the reason why stored pieces of data have similar appearance and characteristics.

In programming, a bit is anything that has two values (often represented by ones and zeros). You can reduce any piece of data into a series of ones and zeros. This process is known as data-to-bit conversion.

Values

Typical computer systems involve billions of bits. That means working with data can be extremely confusing. If you want to work with such a staggering amount of information, you have to divide it into smaller chunks. In JavaScript, these chunks of information are known as values. Although all values are formed by bits, they have different types and functions. JavaScript supports six types of values, namely: strings, numbers, functions, objects, undefined values, and Booleans.

To generate a value, you just have to indicate its name. This process is easy and simple. You won't have to collect materials or pay any fee. Just indicate a value's name, and BAM! You have it.

This chapter will introduce the basic elements of the JavaScript language (i.e. operators and value types). These elements can interact with the values in JavaScript.

For now, let's talk about the different types of values.

The Numbers

Values that belong to this type are… you guessed it right: numeric values. You have to use the following format when writing a JavaScript application:

<number>

For example, let's say you entered this value into your JavaScript program: 99. This will create a bit sequence for the number 99 inside your machine's memory.

The JavaScript language uses 64 bits to store each number value. The number of patterns you can create with 64 bits is limited, so the numbers that you can represent with them is also limited. For x decimal numbers, the numbers that you can represent is equal to 10^x. In the same manner, since you have 64 digits, you may represent up to 2^{64} number values.

Before, computer memory was so small people had to use sets of eight or sixteen bits to indicate number values. Since the memory was insufficient, users often experienced "data overflows" when working with JavaScript. Today, however, even basic desktops possess lots of memory. That means you can easily use 64-bit sequences – you won't have to worry about data overflows anymore.

Number values can also store negative integers. Here, one of the bits represent the value's sign (either "+" or "-"). Additionally, JavaScript supports non-whole numbers (i.e. decimal numbers). To accomplish this, some bits store the decimal point's position in the number value.

You can write decimal and fractional numbers using a period. Here's an example:

99.99

For extremely large or extremely small numbers, on the other hand, you may add an "e" to use the scientific notation. Just follow it up with the correct exponent. Here's an example:

$9.999e9 = 9.999 \times 10^9$

Computations with integers (also known as whole numbers) are 100% accurate. Computations with fractions, however, are not. For example, the value of “pi” cannot be expressed accurately by a limited quantity of decimal numbers. A lot of number values lose their accuracy because there are only 64 bits to keep them. If you want to avoid problems related to fractions, you have to remember that these numbers are just estimates. Don’t treat them as accurate values.

Arithmetic Operations

This is what you do with numeric values. In general, arithmetic operations (e.g. addition, division, subtraction, etc.) use two numeric values to generate a new value. Here’s a sample arithmetic operation written in the JavaScript language:

```
10 + 4 * 5
```

The “*” and “+” signs are known as “operators.” The “+” sign represents addition while “*” represents multiplication. You can generate a new numeric value by placing an operator between 2 numeric values. It is important to note that arithmetic operations in JavaScript follow operator precedence (also called order of operations). Here, you should perform operations in this order: parentheses, exponents, multiplications/divisions, and additions/subtractions. If you will apply operator precedence in the example given above, 10 will be added to the product of 4 * 5.

You can use parentheses to modify the order of operations. For example:

```
(10 + 4) * 5
```

In the JavaScript language, the “-” sign represents subtraction while “/” represents division. You don’t have to worry about operator precedence. If you aren’t sure, however, you can simply add parentheses in your codes.

The Special Numbers

When using JavaScript, you'll encounter three values that are referred to as numbers even if they don't behave as such. These special values are: Infinity, -Infinity, and NaN. – Infinity and Infinity represent negative and positive infinities. In general, you shouldn't trust computations based on infinities. These calculations aren't mathematically reliable.

NaN, on the other hand, means "not a number." You'll get this special value if you will perform arithmetic operations that give imprecise or meaningless result. For example, you'll get NaN if you will divide zero by zero (i.e. 0 / 0).

Strings

You should use strings to represent text-based information. You should use quotes (i.e. either single or double quotes) to enclose your strings. Check the following examples:

'Five little monkeys'

"I'm riding a bicycle"

When creating strings, make sure that you are consistent with your quotes. If you started a line using double quotes, end the line using that same symbol.

In general, you can use quotes to enclose almost anything. Put something between a pair of quotes and the JavaScript language will turn it into a usable string value.

How to "Escape" a Character

There are times when you have to add special characters in your statements. For example, you have to add "newlines" (i.e. the characters you'll get after pressing the Enter key) when writing text strings. Currently, JavaScript requires programmers to add a backslash into quoted texts. By placing a backslash, you are telling the computer that the next character has a special function or meaning. This process is known as "escaping a character."

Quotes that come after a backslash will be considered as a part of the line: they won't

terminate the string. If you'll place "n" after the backslash, a newline character will be added to the statement. Likewise, a "t" that is introduced by a backslash will result to a "tab" character (i.e. the one you'll get after pressing the Tab key). Here is a simple example:

"I'm the first one\nYou are the last"

If you will execute the string given above, you'll get the following:

I'm the first one

You are the last

In certain situations, you want to use an "ordinary" backslash in your codes. You don't want it to be a special tool. You can do this by placing a backslash before another backslash. That is, you are escaping a backslash so you can turn it into a normal character. Analyze the following example:

"You should use \t to add a tab character in your statements"

Once compiled and executed, that line will give you:

You should use \t to add a tab character in your statements

Strings aren't numbers so you can't perform mathematical operations on them. However, you can use the "+" on your strings to concatenate (i.e. combine) them. Here's an example:

"Jav" + "aSc" + "ript"

Because of the "+" signs, the segmented line given above produces: *JavaScript*.

The Unary Operators

In this section, you'll learn about the non-symbol operators. To start the discussion, let's talk about an operator called "typeof." This operator can help you in identifying the type

of any value. Check the following examples:

```
console.log(typeof 4.5)
// → number
console.log(typeof "x")
// → string
```

The sample code given above used console.log to show the results you'll get with "typeof." Whenever you run this kind of code, you should immediately see the value on your computer screen.

The operators given in the previous section required two values, while typeof requires one. Operators that require two values are known as binary operators. Those that work with a single value, however, are known as unary operators.

The Boolean Values

Usually, programmers need values that can distinguish between two options, such as "on" or "off" and "yes" or "no." For this purpose, the JavaScript language supports a value type called Boolean (which has two values only: true and false).

Comparisons

You can use the following method to create Boolean values:

```
console.log(3 > 2)
// → true
console.log(3 < 2)
// → false
```

The < and > characters are the usual symbols used to represent "is less than" and "is greater than", respectively. Both of these symbols fall under the "binary operators" category. If you'll apply them in your codes, you will get a Boolean value that shows whether they are true. You can compare strings using the same method.

```
console.log("Aardvark" < "Zoroaster")
// → true
```

Strings are ordered alphabetically. Also, lowercase letters are “more important” than uppercase ones. That means “a” > “b” is true. JavaScript also includes special characters (e.g. -, !, ?, etc.) in arranging strings. The process of comparison follows Unicode – it assigns a number to each character you might need in programming. This functionality is extremely useful since it allows you to store strings by converting them into number sequences. While comparing string values, JavaScript works from left to right, it checks the number sequences of each character individually.

You may also use similar operators such as <= (i.e. less than or equal to), >= (i.e. greater than or equal to), != (i.e. not equal to), and == (i.e. equal to). Check the following example:

```
console.log("Itchy" != "Scratchy")
// → true
```

In JavaScript, there is just one value that is unequal to itself. This value is NaN or “not a number.”

```
console.log(NaN == NaN)
// → false
```

Originally, NaN is used to represent the outcome of an irrational calculation. As such, it cannot be equal to the outcome of other irrational calculations.

The Logical Operators

In this section, you’ll learn about the different operations that you can apply to a Boolean value itself. The JavaScript language offers three operators, namely: *and*, *not*, and *or*. You can use these operators to perform logical processes on Boolean values. Because of that, they are known as “logical operators.”

a) and

The && symbol represents “and.” This operator is binary. Additionally, it provides a

“true” result ONLY IF the two values you are working on are both true. Here are some examples:

```
console.log(true && false)
// → false
console.log(true && true)
// → true
```

b) or

You should use the “||” symbol to if you want to use the “or” operator. “Or” will give you a “true” result if one of the values you are working on is true. Check the image below:

```
console.log(false || true)
// → true
console.log(false || false)
// → false
```

c) not

You must use “!” (i.e. an exclamation mark) to indicate the “not” operator. This logical operator is unary: you can use it on a single value. However, “not” is unique in that it reverses the value you’ll assign to it. For example, *!false* will give you true while *!true* will give you false.

The Ternary Operator

This logical operator works on three different values. Known as the conditional operator, it requires you to use “?” and “:” in your scripts. The following code will show you how to use it:

```
console.log(true ? 1 : 2);
// → 1
console.log(false ? 1 : 2);
// → 2
```

The value placed on the left-hand side of the “?” determines which of the remaining values will be selected. If it is “true,” the first option is selected. If it is false, however, the

second option is chosen.

Undefined Values

The JavaScript language supports two special values to indicate the lack of a useful value. These special values are “undefined” and “null.” Although these two are considered as legitimate values, they don’t carry any information.

Often, if a JavaScript operation cannot generate a useful value, it gives out “undefined” just because it is required to generate a value.

According to experts, the difference between null and undefined is a mistake on the part of JavaScript designers. That means these two values are considered equal most of the time. If you have to work on these values, just think of them as interchangeable elements in your scripts.

How to Convert Types Automatically

You probably know that JavaScript accepts any application, even those that perform unusual things. The expressions listed below will demonstrate this concept clearly:

```
console.log(8 * null)
// → 0
console.log("5" - 1)
// → 4
console.log("5" + 1)
// → 51
console.log("five" * 2)
// → NaN
console.log(false == 0)
// → true
```

If you will apply an operator to an incorrect value type, JavaScript will automatically convert the said value into the proper type. JavaScript performs this function using a collection of rules that are sometimes unexpected or undesirable. Programmers refer to this automatic process as “type coercion.” Because of this, the null in the first line is converted to 0, while the “5” in the next line is converted to 5 (i.e. string-to-number

conversion). The third line, however, tries to concatenate the values using the “+” sign, so the 1 becomes “1” (number-to-string-conversion).

If you’ll convert a non-number value (“six” or undefined) into a number, you’ll get NaN as a result. Applying mathematical operations on a NaN value will produce more NaNs, so if you are getting this value in unexpected or undesirable situations, check your codes for unwanted type conversions.

Chapter 3: The Program Structure in JavaScript

This chapter will teach you “actual programming.” You’ll be using JavaScript commands that go beyond the nouns and phrases you’ve used so far. Study this material carefully since it will help you master the basics of JavaScript in two weeks.

Statements and Expressions

In the previous chapter, you generated new values by applying operators on existing values. It seems like value creation is the most important part of JavaScript programs. However, it’s just a small part of this powerful language.

The part of a code that creates a value is known as “expression.” Thus, any value that you’ll write literally (e.g. 99 or “ninety-nine”) falls under this category. Expressions are still called expressions even if you’ll enclose them in parentheses. Also, an expression is still an expression even if you are using unary, binary, or ternary operators.

The paragraph above highlights the advantages of a programming interface that is based on language. Expressions act as subsentences in any human language – they can hold other expressions inside them. That means you can mix JavaScript expressions to state confusing calculations.

If expressions are equivalent to subsentences, JavaScript statements are equivalent to complete sentences. Basically, a JavaScript program is just a collection of statements.

The most basic type of statement is a value followed by a semicolon. Based on this principle, we can create this simple program:

```
0;

false;
```

In JavaScript, the sample given above is considered as a complete program. It’s a

worthless one, though. Statements are only useful if they do something that affects other things (e.g. showing some characters on the screen, improving the programming environment, etc.). The two statements shown above simply create two values (i.e. 0 and false). They don't change or improve anything. If you'll execute this program, you won't observe anything significant.

The Variables

How can JavaScript programs maintain themselves? How do they store things? You have seen how to create new values using old ones. However, that process doesn't modify the preexisting values. You also need to use the resulting values immediately or they will simply disappear. To capture and hold values, JavaScript offers variables.

*var example = 3 * 3;*

In this example, var (one of the keywords in JavaScript) indicates that the line will create a variable. The keyword is followed by the variable's name and, if you want to assign a value immediately, by an "=" sign and your desired expression.

The statement generates a variable named "example" and uses it to store the product of 3 * 3.

Once you have defined a variable, you can use its name as an ordinary expression in your future JavaScript statements. The value that you'll get from that "expression" is identical to the value you assigned while creating it. Check the following example:

```
var three = 3;

console.log(three * three);

// -> 9
```

When naming variables, you can use any word that isn't a keyword (e.g. var). You can include numbers in the variable's name, but you can't use a number as the initial character.

For example, *programming1* is valid but *1programming* isn't. Additionally, you cannot use spaces in naming variables. In terms of special characters, you can only use the dollar sign (i.e. $) and the underscore (i.e. _). JavaScript doesn't allow you to use punctuation marks in variable names.

You may change the value assigned to a variable. In JavaScript, you can simply use the "=" sign on any variable to give it a new value.

```
var taste = "bitter";

console.log(taste);

// -> bitter

taste = "sweet";

console.log(taste);

// -> sweet
```

In general, you should think of variables as tentacles instead of containers. A variable grasps values, it doesn't contain them. That means multiple variables can point to a single value.

Let's analyze a simple example. You need to generate a variable to remember the amount Johnny owes you. When he pays you $50, you have to assign a new value to the existing variable. Check the lines below:

```
var JohnnysDebt = 200;

JohnnysDebt = JohnnysDebt – 50;

console.log(JohnnysDebt);

// -> 150
```

If you will create a variable without assigning a value, it will be considered empty. It will

give you “undefined” if you’ll ask for the value it contains.

JavaScript allows you to include several variables in one var statement. When using this functionality, you should separate the variables using commas. Here’s an example:

var three = 3, four = 4;

console.log(three + four);

// -> 7

The Reserved Words in JavaScript

As discussed earlier, you can’t use keywords when naming your variables. However, there are other words that you can’t use as variable names because JavaScript has reserved them for future developments. Here are some of the reserved words in this scripting language:

in, do, let, if, new, try, with, case, void, with, while, false, yield, extends, delete, default, const, debugger, class, break, finally, null, private, return, import, protected, switch, super, interface, instanceof, throw, implements

Keep in mind that you can’t use the words given above when defining variables. If you’ll forget about this simple rule, you might experience problems when working with JavaScript programs.

JavaScript Environments

The term “environment” refers to the set of active values and variables. Basically, environments cannot be empty while starting up a JavaScript program. Environments hold variables related to the language itself, and often, they have variables that allow interactions with computer systems. For instance, in web browsers, various functions and variables exist to check and affect the current website and read the inputs from the mouse and keyboard.

The Functions

The default JavaScript environment involves functions. Functions are pieces of computer programs linked to a value. You may apply these values to activate the linked program. For instance, in browser environments, alert (i.e. a variable) contains a function that displays a dialog box. Here's an example:

alert("You are not authorized to access this page!");

You may execute a function by placing parentheses after a JavaScript expression that generates a function value. Often, you'll indicate the name of the variables that contain your desired functions. By enclosing values inside the parentheses, you are assigning those values to the program/s within the function. For the previous example, "alert" uses the assigned string to show a message in the resulting dialog box. Programmers use the term "arguments" when referring to the value/s assigned to a function.

The "console.log"

You can use "alert" when trying out new strings or statements. However, since it requires you to close out the resulting windows, you might find it inconvenient and time-consuming. Because of this, you might want to use the function you've seen in the earlier sections of this book: console.log. Almost all JavaScript systems (e.g. Node.js, web browsers, etc.) offer this function to transfer arguments to a text-compatible output device.

When it comes to web browsers, the code's output goes to the console of JavaScript. Although this section of the browser GUI is usually hidden, you can access it by pressing certain keys (i.e. for Windows computers, hit F12; For Mac computers, hit Command-Option-I). If you are having problems viewing this part, you may run an intra-device search for "developer tools" or "web console." Analyze the following example:

var a = 99;

console.log(“one hundred minus one is”, a);

// -> one hundred minus one is 99

As discussed earlier, you can’t use periods or commas when naming variables. However, the example given above contains one. The reason for this is that console.log isn’t an ordinary variable. It is a JavaScript expression that collects the “log” information from the value stored in “console” (which is a variable).

The Return Values

Writing information on the screen and displaying dialog boxes are known as side effects. Because of these side effects, functions are important in the JavaScript language. You may also use functions to generate values – in this case, a function can be useful even without its side effects. For instance, Math.max (a JavaScript function) receives an unlimited number of numeric values and indicates the highest one. Here’s an example:

console.log(Math.max(1, 2, 3, 4, 5));

// -> 5

In JavaScript, anything that generates values is considered as an expression. That means you can perform function calls in large expressions. In the example below, you’ll use the Math.min function as an input for the “+” operator:

console.log(Math.min(1, 2) + 98);

// -> 99

Chapter 4: More Information About Functions in JavaScript

In the previous section, you've learned how to create and call function values. This chapter will help you understand this important topic further. According to expert programmers, functions serve as your main tools when using JavaScript. Functions are powerful tools that can help you design large programs, minimize redundancy, name subprograms, and secure isolation for each subprogram. That means you have to master functions if you want to be use JavaScript effectively. This chapter will help you to learn this scripting language in just two weeks.

Many people apply functions to define new vocabularies. That means you can use them to create words and assign meanings.

An average English speaker has about 20,000 words in his/her vocabulary. Well, programming languages usually don't have that amount of built-in commands. That means the vocabulary in programming is more precise than those used in human languages. If you want to minimize redundancy, you have to add your own words into JavaScript and create your own vocabulary.

How to Define a Function

Defining a function is like defining an ordinary variable. The only difference is that the value you are assigning is a function. For instance, the code below defines a variable named "square." This variable points to a function that generates the square of any assigned number.

```
var square = function(x) {
  return x * x;
};

console.log(square(12));
// → 144
```

When creating a function, you should start an expression with this keyword: “function.” Each function has two parts: (1) body and (2) parameter/s. The function’s body stores the JavaScript statements that you want to execute once the function is invoked (or called). Additionally, you should always enclose the body using curly braces (even if it contains a single statement).

A parameter, on the other hand, is a name listed in the function’s definition. Functions may have one or more parameters. Actually, you can even create a function without entering any parameter. To help you understand this concept, let’s create two functions: makeNoise and power.

```
var makeNoise = function() {
  console.log("Pling!");
};

makeNoise();
// → Pling!

var power = function(base, exponent) {
  var result = 1;
  for (var count = 0; count < exponent; count++)
    result *= base;
  return result;
};

console.log(power(2, 10));
// → 1024
```

As you can see, makeNoise doesn’t have any parameter. Meanwhile, “power” has two (i.e. base and exponent).

The Parameters and Scopes in JavaScript

A function’s parameters behave like ordinary variables, although they get their original value from the programmer, not from the function they are linked to.

Here is an important property of any function: all of the variables created in it (even the parameters) are local to it. That means, for instance, that “result” in the power statement will refresh itself each time the function is invoked. These separate repetitions don’t affect

each other.

It is important to note that this “principle of locality” is applicable only for variables and parameters defined using “var.” Additionally, you should define the variables and parameters within the function’s body. A variable defined outside of a function is called global, since it can be seen and accessed throughout the JavaScript application. You may access global variables and parameters from within a function if they have no “local” counterparts.

The code below illustrates this concept. It declares and invokes two functions that give a value to “x.” The first function defines the variable as a local one. The second function, on the other hand, doesn’t. That means if you will reference x inside the function, you’ll point to the global x (shown at the first line of the example).

```
var x = "outside";

var f1 = function() {
  var x = "inside f1";
};
f1();
console.log(x);
// → outside

var f2 = function() {
  x = "inside f2";
};
f2();
console.log(x);
// → inside f2
```

This functionality helps you in preventing unintended references or interactions between different functions. If JavaScript allows you to share all variables throughout the entire program, you’ll have to spend lots of time and effort in ensuring that names aren’t duplicated. Reusing variable names can give unexpected or undesirable effects on your codes. Thus, it can ruin the programs you are working on.

Since this language treats local variables as if they exist only inside their respective

function, it can work on functions as distinct worlds. JavaScript won't have to worry about the entire application whenever you work on a single function.

The Nested Scope

Aside from defining local and global variables, JavaScript uses other methods to distinguish different kinds of functions. For example, you may create functions within existing functions. This technique produces multiple levels of locality.

For instance, the function given below contains two functions:

```
var landscape = function() {
  var result = "";
  var flat = function(size) {
    for (var count = 0; count < size; count++)
      result += "_";
  };
  var mountain = function(size) {
    result += "/";
    for (var count = 0; count < size; count++)
      result += "'";
    result += "\\";
  };

  flat(3);
  mountain(4);
  flat(6);
  mountain(1);
  flat(1);
  return result;
};

console.log(landscape());
// → ___/''''\______/'\_
```

Both functions (i.e. flat and mountain) can access the "result" variable because they are placed within the function that declares that variable. However, they can't access each other's variables. That means the program environment located outside the function cannot access the variables it contains.

Programmers who have used other languages might think that they can create new local environments by enclosing code blocks using curly braces. In JavaScript, however, only functions can generate new scopes.

Using Functions as Ordinary Values

Usually, function variables serve as names for a certain piece of a program. You cannot modify existing function variables: once you have defined them, they will stop you from performing any future modifications. This characteristic can cause confusions regarding names and the functions they refer to.

Obviously, a function and a variable are two different things. Function values can do what typical values can do – you may call and use them for complex expressions. You may change the location of a function value, assign it as an argument for a function, etc. In the same manner, variables that contain functions are still considered as regular variables. That means you can give them new values.

Declarations

JavaScript offers a quick and easy way to declare functions. You just have to start your statements using the keyword “function.” For instance:

```
function square(x) {
  return x * x;
}
```

The statement given above is known as a “declaration.” It defines a variable named square and gives it a certain function. This example is easy to understand. However, this method of defining functions has a subtlety:

```
console.log("The future says:", future());

function future() {
  return "We STILL have no flying cars.";
}
```

The latest example works perfectly, even if the function is declared under the code it is assigned to. This code is valid because declarations ignore the top-to-bottom statement structure. You can actually move a declaration onto the top of its scope so that local codes can access it. This functionality can be useful since it allows you to arrange codes in a meaningful way, without having to define all of the affected functions.

The Optional Arguments

The JavaScript language accepts the code given below. You won’t encounter any problem while invoking it.

alert(“Hi”, “Good Morning”, “How are you?”);

Officially, the “alert” function accepts a single argument only. However, it won’t complain if you’ll invoke it using the sample code above. It will just accept the first argument (i.e. “Hi”) and ignore the rest.

This programming language is “open-minded” when it comes to the quantity of arguments you can assign to your functions. If you’ll assign too many arguments, it will accept the valid ones and ignore the others. If your arguments are not enough, however, JavaScript will complete your statement by assigning “undefined” to the appropriate parameters.

Python Academy

The Stress Free Way To Learning Python Inside & Out

Table Of Contents

Introduction

I want to thank you and congratulate you for downloading the book, *" Learn the Basics of Python Programming in 2 Weeks "* .

This book contains proven steps and strategies on how to write and run programs in Python. There are sample codes and programs that you can use as guidelines.

This book also contains an introduction to the programming language, including a brief history and the fundamentals of Python. It also contains detailed explanations about the features found in Python.

Thanks again for downloading this book, I hope you enjoy it!

This document is geared towards providing exact and reliable information in regards to the topic and issue covered. The publication is sold with the idea that the publisher is not required to render accounting, officially permitted, or otherwise, qualified services. If advice is necessary, legal or professional, a practiced individual in the profession should be ordered.

- From a Declaration of Principles which was accepted and approved equally by a Committee of the American Bar Association and a Committee of Publishers and Associations.

The information provided herein is stated to be truthful and consistent, in that any liability, in terms of inattention or otherwise, by any usage or abuse of any policies, processes, or directions contained within is the solitary and utter responsibility of the recipient reader. Under no circumstances will any legal responsibility or blame be held against the publisher for any reparation, damages, or monetary loss due to the information herein, either directly or indirectly.

Chapter 1: Introduction to Python

If you are looking for a general purpose, high level programming language with code readability as its main focal point, you should consider Python. It has a syntax that allows you to express concepts using fewer lines of code than other programming languages, such as C, C++, and Java. It supports a wide range of programming paradigms, thereby encompassing imperative, functional and object-oriented programming. Also, it features a comprehensive standard library, a dynamic tape system and an automatic memory management.

There are plenty of interpreters you can use for installation on different operating systems; hence, it is possible for you to execute Python on various systems. You can even make use of third-party applications. If you do not want to install an interpreter, you can package the code into standalone executable programs so that you can effectively distribute Python-based software to different environments.

A Brief History of the Python Programming Language

Python was developed by Guido van Rossum in the late 1980 ' s. It was initially just a Christmas project. Van Rossum wanted an interpreter for a scripting language that C and UNIX hackers will use. His little project eventually upgraded to Python 2.0, which was released on October 2000. This new version had a complete garbage collector and Unicode support. Python 3.0, also called Python 3000 and py3k, was released in December 2008 and had features that were backported to versions 2.6 and 2.7.

Why should you Use Python?

Programming languages exist for a reason. Python, for instance, was developed to allow programmers to create efficient and productive codes. Its main objective is to help beginners learn a programming language easily and quickly. Due to this less learning time, you can create more useful applications that will be difficult to do with more obscure and complicated programming languages.

With Python, you can also benefit from less development time when coding applications. As mentioned earlier, Python has fewer lines of code than C, C++, and Java. Its codes are actually five to ten times shorter; thus, making it more efficient and less complicated. You get to spend less time in developing applications and more time tweaking and improving them.

When it comes to checking for bugs and errors, it is crucial for the programming language that you use to be easy to read and comprehend. If the programming language is too complicated, you may have a hard time coding and checking your program. With Python, codes are much easier to read and write; hence, you can easily interpret the codes and make the necessary changes.

Furthermore, Python has many other uses. It is ideal for scripting applications that are browser-based, creating great user interfaces and rough application examples, interacting with databases, working with XML and designing mathematic, engineering and scientific applications.

Python vs. C#, Java, and Perl

You may find comparing programming languages with one another to be a subjective task. Usually, their differences are a matter of personal preference. Nonetheless, there are times when such comparisons are backed up by scientific data. Anyway, you have to keep in

mind that an all-encompassing programming language does not exist. As a programmer, you just have to find one that works best for your goals or needs.

C#

If you have a background in Java, you may notice that C# and Java are highly similar. Then again, C# still has its own advantages and disadvantages compared to Java. Microsoft claims that their primary objective in developing C# is to produce a better version of C and C++. Compared to C#, however, Python has better scientific and engineering applications and better multiplatform support. It is more extendable when it comes to using C, C++ and Java. It is easier to learn and comprehend, and it allows the use of various computer environments. It also has more concise codes.

Java

Programmers consider Java as a one-stop shop programming language. For many years, they have searched for something that can be run and written anywhere and they found Java, which can perform well in different platforms and computer environments. With this being said, Python is also a one-stop shop programming language. It is very similar to Java, except that it has more concise codes and it is easier to learn and comprehend. It is much faster to develop and it has improved data boxes or variables that can store different data types that are useful when running applications.

PERL

PERL stands for Practical Extraction and Report Language. It is a programming language that is suitable for acquiring and processing data from another database. In comparison, Python is better than PERL because it is easier to read, implement, learn and understand. It has better Java integration and data protection. It also has less platform-specific biases.

Why Python is Ideal for Beginners?

If you have just started programming, you may want to consider Python. It is ideal for beginners because it has a consistent and simple syntax and vast standard library that allows you to do multiple projects. The assignments involved are not limited to the usual four-function calculator and check balancing programs.

As you get used to writing programs in Python, you will realize that it is actually easy to make realistic applications. The interactive interpreter will also allow you to test language features. In Python, you can concentrate on essential programming skills, such as programming decomposition and data type design, and fundamental concepts, including procedures and loops.

Since Python involves the use of multiple libraries and system calls, you can develop applications with interfaces that are easy to program. You can also complete tasks necessary for the application programming interface (API).

Do not worry if you have never used any other programming language before. Even people with no prior programming knowledge or experience can easily grasp the fundamentals of the Python programming language.

As for the installation, Python is easy to install. Most UNIX and Linux distributions actually include it in their package. If you purchase a Windows computer from Hewlett-Packard (HP), you can readily use Python as it comes pre-installed with the system.

To make things easier for you, you should study how to use the text editors as well as the integrated development environments (IDEs). It will also be helpful to read programming books with sample codes and programs.

Regarding copyright, the developers of Python allow programmers to do whatever they want with the source, as long as they do not forget to include the copyrights. The copyright rules are not that strict. You can even sell copies in binary and source form, as well as products involving Python use. However, if you wish to use the logo, see to it that

you obtain permission.

Python is highly stable. In fact, it is stable enough for regular use. You can expect a new version within six to eighteen months. The developers issue bug fix releases to ensure that the newer versions are better than the previous ones.

If you want to perform a static analysis or search for bugs, you can use Pylint or PyChecker. The previous is a tool that checks the module to see if it abides by the coding standard as well as allow the customization of plug-ins. The latter is a static analysis tool that finds bugs in the source code.

So now that you have learned about the fundamentals of the programming language, you may still wonder how Python got its name. Was Guido van Rossum fond of pythons? Well, he was actually fond of the television show called Monty Python ’ s Flying Circus, not the reptile.

During the time of Python ’ s development, he was reading scripts from the comedy series and thought that ‘ Python ’ will be a suitable name since it was short, unique and has the right amount of mystery. In fact, references to the comedy show are allowed and actually encouraged in documentations.

Chapter 2: Syntax

Python has a simple and straightforward syntax. It even encourages programmers to write programs without using prepared or boilerplate codes. The print directive is actually the simplest of all directives. It prints out lines and includes newlines. You may notice that the print directive is different in the new major versions of the programming language.

Python 2.0 is the more common version while Python 3.0 supports the latest features and is more semantically correct. Anyway, the print statement is not considered as a function in version 2.0; hence, you can invoke it without including parentheses in your code. On the other hand, the print statement is considered as a function in version 3.0; hence, you have to use parentheses if you wish to invoke it.

Interactive Mode Programming

You can execute your programs in different modes. If you invoke the interpreter without passing the script file as a parameter, this is what you will get:

```
$ python
Python 2.4.3 ( #1, Nov 11 2010, 13:34:43 )
[GCC 4.1.2 20080704 ( Red Hat 4.1.2 – 48 )] on linux2
Type “ help ” , “ copyright ” , “ credits ” or “ license ” for more information.
>>>
```

When you see this prompt, you can type in your desired text then press Enter. In this example, we will be using the words ‘ Monty Python and the Holy Grail ’ .

```
>>> print “ Monty Python and the Holy Grail ” ;
```

Take note that if you are using a newer version of the programming language, you need to use opening and closing parentheses with your print statement, such as in the following:

```
>> print ( “ Monty Python and the Holy Grail ” ) ;
```

Regardless of which version you are using, if you run the sample code shown above, you will get the following output:

```
Monty Python and the Holy Grail
```

Script Mode Programming

If you invoke the interpreter with a script parameter, the script will start to execute and continue to run until it is done. When it is done, the interpreter will not be active anymore. Consider the following example. The sample program is written in a script and has a *.py* extension:

```
print “ Monty Python ’ s Flying Circus ” ;
```

If you type in the above given source code in a test.py file and run it as

```
$ python test. py
```

you will obtain the following output:

```
Monty Python ’ s Flying Circus
```

Another way to execute scripts is to modify the *.py* file, such as:

```
#! /usr /bin /python
```

```
print “ Monty Python ’ s Flying Circus ” ;
```

If you run it as

```
$ chmod + x test.py
$ ./test.py
```

you get the following output:

```
Monty Python ’ s Flying Circus
```

Identifiers

An identifier is basically used to determine functions, variables, modules, classes, and any other objects. It begins with an underscore (_) or a letter. It is then followed by digits, underscores, zero or other letters. As a programmer, feel free to use any letter or digit. You can use uppercase and lowercase letters.

However, you cannot use punctuations and special characters, such as @, $, and %, within the identifiers. In addition, Python is a case sensitive programming language. This means that you have to be careful when you use uppercase and lowercase letters in your codes. For instance, *wendy*, *Wendy*, and *WENDY* are all the same name and yet they are regarded as three different identifiers in Python.

Rules for Identifiers in Python

There are several rules that you have to abide by when writing programs in Python:

The class name must always start with an uppercase character while the rest of the identifiers must start with a lowercase character.

The identifier is private if it starts with just one leading underscore.

The identifier is strongly private if it starts with two leading underscores.

The identifier is a language-defined special name if it ends with two trailing underscores.

Reserved Words

Take note that there are certain words you cannot use as constants, identifier names, or variables in Python. All keywords are also written using lowercase letters. The following is a table of the reserved words in the programming language:

And	Assert	Break	Class	Continue
def	del	elif	else	except
exec	finally	for	from	global
if	import	in	is	lambda
Not	or	pass	print	raise
return	try	while	with	yield

Indentation and Lines

There are no braces for the indication of blocks of code for class definition and function in Python. Likewise, flow control is not included. If you want to denote such blocks of code, you have to make use of line indentation. You can adjust it for spaces, but make sure to indent all statements with a block, too. To help you understand this further, consider the following sample codes:

```
if True:
    print “ Correct ”
else:
    print “ Incorrect ”
```

```
if True
    print “ Input ”
    print “ Correct ”
else:
    print “ Input ”
    print “ False
```

Running the first given example generates an output. Running the second one, however, results in an error. Why did this happen? Well, you have to keep in mind that in Python, blocks are formed by indenting continuous lines with the same amount of space.

Indentation is simply a way to group statements. Programmers use it in place of curly

braces of blocks of code. Tabs and spaces are supported, but standard indentation requires standard codes to have similar amounts of spaces. In general, four spaces are used. Take a look at the following example:

```
w = 1

if w == 1 :

    # This shows an indentation with exactly four spaces

    print “ w is 1 . “
```

Indentation Myths

There are certain myths that surround indentation in Python. Here are some of them:

A whitespace is necessary in every source code.

Actually, you do not have to use whitespaces in all your source codes. A whitespace is not necessarily significant, although an indentation is. As you have learned, this is the whitespace found at the very left part of a statement. Everywhere else, a whitespace is not that significant and may be omitted. You can use it in any way you like, even inserting arbitrary whitespaces or empty lines that do not contain anything anywhere in your program.

Moreover, the exact amount of indentation does not really matter, but the relative indentation of your nested blocks does. The indentation level is actually not recognized when you use implicit or explicit continuation lines. For instance, you may split a list across multiple lines. The indentation is just not significant at all. Take a look at the following example:

```
foo = [
        ‘ a string ‘ ,
        ‘ another string ‘ ,
        ‘ a short string ‘
]
print foo
```

If you run the above given code, you will get the following output:

```
[ ‘ a string ‘ , ‘ another string ‘ , ‘ a short string ‘ ]
```

Here is another example:

```
bar = ‘ look at this example ‘ \
    ‘ of a long string ‘ \
            ‘ that is split ‘ \
    ‘ across multiple lines ‘
print bar
```

If you run the above given code, you will obtain the following output:

```
look at this example of a long string that is split across multiple lines
```

A certain style of indentation should be used in your programs.

Well, this one is both true and untrue. You can write the inner block on a line and not indent it. You can use any of the following versions of the “*if statement*” since

all of them are valid and produce the same output:

```
if 1 + 1 == 2 :
    print “ foo”
    print “ bar ”
    w = 99
```

```
if 1 + 1 == 2 :
    print “foo” ; print “ bar “ ; w = 99
```

```
if 1 + 1 == 2 : print “ foo “ ; print “ bar “ ; w = 99
```

As a programmer, you may wish to write your block of code in separate lines, such as the one shown in the first example. However, there are times when there are similar *if statements* that you can conveniently write on each line.

In case you decide to write your block of code on separate lines, then you have to follow the rules of indentation. You have to indent the enclosed block more than the “*if statement*”.

In conclusion, you will be forced to abide by this rule in Python, unless you opted to make the structure of your program more complicated. The programming

language does not allow program structure obfuscation with the use of fake indentations.

Keep in mind that blocks are denoted by indentation in the Python programming language; thus, the indentation is the same in every program. The consistency of the code formatting makes the program easier to read and understand.

It is not possible to mix spaces and tabs in Python.

Yes, this one is true, even for programs written in the C language. You cannot mix spaces and tabs safely. Even though there would not really be a huge difference for your compiler, you may have a hard time dealing with codes. For instance, if you move a C source to one editor that has different tab stops, bugs will be easier to introduce.

Once again, keep in mind that it is not ideal to mix spaces and tabs for indentation. You can use spaces or tabs alone, though. In addition, you may want to avoid tabs altogether. After all, the semantics of tabs are not that well-defined and may appear differently on various types of editors and systems.

Tabs are also often wrongly converted or destroyed during copy and paste operations, as well as whenever a source code gets inserted into a Web page or any other type of markup code.

It is possible to skip the indentation and use a keyword instead.

Yes, you can skip using an indentation and just use a keyword. There are actually a few programmers who prefer to use *endif* instead of an indentation to indicate the end of a block of code.

Well, it is not exactly a recognized keyword in Python. The earlier versions of the programming language come with a tool that converts code using the keyword *end* to correct the indentation and remove such keyword.

This may be used as a pre-processor to the compiler. In the recent versions of the programming language, however, the tool has been removed, most probably because it is not often used.

How is the indentation parsed by the compiler?

The parsing is actually simple and well defined. In general, the changes to the level of indentation are inserted as tokens into the stream. The indentation levels are stored using a stack from the lexical analyzer or tokenizer. At first, the stack only has a value of 0, which is found at the leftmost part.

Each time a nested block starts, a new level of indentation gets pushed on the stack. The *indent token* is then inserted into the stream, which is eventually passed on to

the parser. It is not possible to have more than a single indent token in succession.

In the event that a line is seen with a smaller level of indentation, the values start popping from the stack until one of them gets on top. This is equivalent to the new level of indentation. In case there is nothing found, a syntax error is generated. For ever value popped, there is a *dedent token*. It is possible to have multiple dedent tokens in succession. At the end of every source code, there are dedent tokens generated for the level of indentation that is left at the stack. This continues to occur until there is 0 left.

Multiline Statements

When you end a statement, you can either use a new line or a continuation symbol (\ ) if you want to indicate that the line needs to continue. To help you understand this concept further, consider the following example:

```
total = first_item + \
    second_item + \
    third_item
```

There is no need for you to use the continuation symbol when you write statements that are contained within brackets, such as { }, ( ), and [ ]. For instance, if you wish to display the months in a year, you may simply write:

```
year = [ ‘ January ’ , ‘ February ’ , ‘ March ’ , ‘ April ’ , ‘ May ’ , ‘ June ’ , ‘ July ’
```

, ‘ August ’ , ‘ September ’ , ‘ October ’ , ‘ November ’ , ‘ December ’]

You are allowed to write multiple statements on a single line or create multiple groups as suites. When it comes to writing multiple statements, keep in mind that the inclusion of the semicolon (;) is crucial. The semicolon allows you to write as many statements as possible, as long as they do not start a new block of code. Consider the following example:

```
import sys ; y = ‘ bar ’ ;  sys.stdout.write ( y + ‘ \n ’ )
```

So what are suites exactly? Well, they are groups of statements that consist of blocks of code. Compound or complex statements, such as *if*, *while*, *def*, and *class* require a suite and a header line.

So what are header lines? They begin statements with a keyword and end them with a colon (:) . Also, they are followed by one or more lines that make up a suite. Consider the following example:

```
if expression :
    suite
elif expression :
    suite
else :
    suite
```

Quotation

As a programmer, you are allowed to use a single quote (‘) , double quote (“), and a triple quote (‘’’ or “””) when you denote string literals. Then again, see to it that you use the same type of quotes at the start and end of your string. Typically, triple quotes are used to span strings across multiple lines. Take a look at the following example:

```
paragraph = “”” You are reading an example of a paragraph that consists multiple lines and sentences. You are an excellent programmer. “””
```

Comments

When it comes to comments, you should use the hash symbol (#) to start them. However, this hash symbol should not be within a string literal. Also, the characters after it towards the end of the line should be included in the comment. In Python, comments are not recognized by the interpreter. To help you understand this concept further, take a look at the following example:

```
# This is the first comment

print “ Monty Python ’ s Flying Circus is a British sketch comedy series. ” ;

# This is the second comment
```

If you run the example given above, you will obtain the following output:

```
Monty Python ’ s Flying Circus is a British sketch comedy series.
```

You can also write another comment after an expression or a statement, such as in the following:

```
name = “ Wendy ” # This is a sample comment
```

If you want to comment on multiple lines, you may do so as well. For example:

```
# This is a sample comment.

# This one is also a comment.

# This is another comment.

# This comment is written by Wendy.
```

Blank Lines

These lines are not recognized in the Python programming language. With this being said, they are pretty much like comments. They contain whitespaces and even comments. You have to use empty lines to terminate multiline statements in an interactive interpreter session.

Chapter 3: Data Types

In Python, input data are sorted according to different categories. The primary purpose of sorting is to help programmers like you in processing information more efficiently. Such categories function as data storage locations that you can access whenever you run the Python platform.

Variables

Variables contain values that have been specifically allocated to them. If you are working on complex codes for applications, you may want to store your information in these variables. Do not worry because you can access them anytime you need them. You can even use them to ensure that the information you gather from your end users stay safe and secured.

Numeric Types

Numbers in the Python programming language are different from the numbers you use to solve problems in Algebra. In Mathematics, adding *.0* at the end of a number practically means nothing. It does not make any difference to its value. For instance, *3* and *3.0* are the same.

In Python, however, *3* and *3.0* are different numbers. Before the program processes it, it has to undergo certain data processing methods. As a programmer, you have to learn about the different numeric types.

Integers

All whole numbers are integers. Numbers that contain a decimal point are not whole numbers; therefore, 3 is a whole number while 3.0 is not. Integers in Python are characterized by the data type *int*.

Take note that integers have capacity limitations. You will generate an error if you try to process a value beyond the allowed limits. Integers typically process numbers between -9,223,372,036.854 and 9,223,372,036.854.

There are interesting features that come with the *int* variable. For instance, base 2 only uses 0 and 1, base 8 uses numbers from 0 to 7, base 10 has similar properties with the decimal system and base 16 uses the letters A to F and the numbers 0 to 9 as digits.

Floating Point Values

Any number that contains a decimal point is considered as a floating point value in Python. It does not matter if the number after the decimal point is *0* or not. *3.0*, *1.5*, and *11.4*, for example, are all floating point values. They are stored in the float data type. One huge advantage of floating point values over integers is that they have bigger storage spaces; hence, they are capable of storing very small or large values.

Then again, you should not think that they have an unlimited storage space. There is still a limitation. Nevertheless, they can contain as little as $\pm 2.2250738585072014 \times 10^{-308}$ and as much as $1.7976931348623157 \times 10^{-308}$. There are a couple of ways to allocate values with the use of floating point values. First, you can directly assign them. Second, you can use a scientific notation. Keep in mind that negative exponents produce fraction equivalents.

Complex Numbers

These numbers consist of real numbers and imaginary numbers combined. Usually, they are used in dynamic systems, quantum mechanics, computer graphics, electrical engineering and fluid dynamics. Complex numbers can be processed in Python and a few other programming languages.

Boolean Values

These are the two constant objects *True* and *False*. They represent truth values. When used in a numeric context, they function as 0 and 1. You can use the function *bool* to assign a value to a Boolean if such value may be interpreted as a truth value.

Strings

They are groups of characters that are connected with double quotation marks. Consider the following example:

```
TheString = “ Python got its name from a popular comedy show. “
```

As you can see in the sample code shown above, the phrase *Python got its name from a popular comedy show.* is assigned to the variable *TheString*.

Computers cannot understand letters, only numbers. So when you write a program, Python reads and interprets it based on the numbers that represent its letters. For example, in the American Standard Code for Information Interchange (ASCII), the number *65* represents the letter *A*. So if you type in

```
ord ( “ A “ )
```

you will get an output of

```
65
```

Because computers cannot understand letters, you have to convert strings into numbers. You can use *int* or *float* to do this. In case you need to convert numbers into strings, you can use *str*.

Chapter 4: Operators

The values of your operands are manipulated by operators. There are seven types of operators used in the Python programming language. The following tables display these operators and provide brief explanations regarding their function.

Arithmetic Operators

Operator	*Description*
Addition (+)	It adds the values.
Subtraction (-)	It subtracts the second operand from the previous operand.
Multiplication (*)	It multiples the values.
Division (/)	It divides the first operand by the second operand.
Modulus (%)	It divides the first operand by the second operand and returns the remainder
Exponent (**)	It performs exponential calculation on the operators.
Floor Division (//)	It divides the operands but eliminates the decimal points after the result.

Comparison Operators or Relational Operators

Operator	*Description*
==	If the values of both operands are equal, the condition is true.
!=	If the values of both operands are not equal, the condition is true.
<>	If the values of both operands are not equal, the condition is true.
>	If the value of the left operand is bigger than the value of the right operand, the condition is true.
<	If the value of the left operand is less than the value of the right operand, the condition is true.
>=	If the value of the left operand is bigger or equal to the value of the right operand, the condition is true.
<=	If the value of the left operand is less than or equal to the value of the right operand, the condition is true.

Assignment Operators

Operator	*Description*
=	It assigns values from the right operand to the left operand.
+= add AND	It adds the right operand to the left operand, and then allocates the result to the left operand.
-= subtract AND	It subtracts the right operand from the left operand, and then allocates the result to the left operand.
*= multiply AND	It multiples the left operand and the right operand, and then allocates the result to the left operand.
/= divide AND	It divides the left operand with the right operand, and then allocates the result to the left operand.
%= modulus AND	It uses the two operands to find the modulus, and then allocates the result to the left operand.
**= exponent AND	It performs exponential computation on the operators and then assigns the value to the left operand.
//= floor division	It performs floor division on the operators and assigns the value to the left operand.

Bitwise Operators

Operator	*Description*
& binary AND	It copies the bit if it is present in both operands.
| binary OR	It copies the bit if it is present in either operand.
^ binary XOR	It copies the bit if it is present in one operand, but not both.
~ binary ones complement	It flips bits.
<< binary left shift	It moves the value of the left operand towards the left based on the number of bits assigned by the right operand.
>> binary right shift	It moves the value of the left operand towards the right based on the number of bits assigned by the right operand.

Logical Operators

Operator	*Description*
And logical AND	The condition is true if both operands are true.
Or logical OR	The condition is true if an operand is non-zero.
Not logical NOT	It reverses the logical state of the operand.

Membership Operators

Operator	*Description*
Is	If the variables on either side of the operator point toward the same object, it evaluates to true. Otherwise, it evaluates to false.
Not in	If it does not find a variable in a particular sequence, it evaluates to true. Otherwise, it evaluates to false.

Identity Operators

Operator	*Description*
Is	If the variables on either side of the operator point towards the same object, it evaluates to true. Otherwise, it evaluates to false.
Is not	If the variables on either side of the operator point towards the same object, it evaluates to false. Otherwise, it evaluates to true.

Conclusion

Thank you again for downloading this book!

I hope this book was able to help you learn about the Python programming language.

The next step is to apply what you have learned from this book.

Finally, if you enjoyed this book, then I ’ d like to ask you for a favor, would you be kind enough to leave a review for this book on Amazon? It ’ d be greatly appreciated!

Click here to leave a review for this book on Amazon!

Thank you and good luck!

WINDOWS 10 BOOTCAMP

Learn the Basics of Windows 10 in 2 Weeks!

Table Of Contents

Introduction

I want to thank you and congratulate you for downloading the book, *"Learn the Basics of Windows 10 in 2 Weeks!"*

Being the newest incarnation of the Windows Operating System Franchise, Windows 10 is slowly taking over the world—which is exactly why you should learn how to use it, and make sure that it works for your benefit!

This book contains proven steps and strategies on how to use Windows 10—from how to set it up, to how various parts of it are used—and you'll be able to learn all these in just a matter of 2 weeks!

In this book you will learn:

How to install Windows 10

How to use the Start menu

How to use the features and settings

And much more!

Read this book now to find out how!

Thanks again for downloading this book, I hope you enjoy it!

Before we start…Bonus: Receive More Free Books By Me!

Thanks for reading my book! I hope you received awesome value from it. I'd like to thank you by giving you more free books, and info related to programming and technology!

Simply click on the link below and enter your email address on my website!

Everything you will get will be useful to you if you're serious about changing your programming skills. I will be offering:

More Books By Me

How I was able to learn programming in record time

Info On My Favourite Products Related To Programming

And Much More!

Click Here to get access to all this info!

Part 1: First Week

Here's what you can learn in the first week of installing Windows 10!

Chapter 1: Installing Windows 10

Before installing Windows 10, you have to make sure that your computer is ready, which means you have to follow the following steps below:

The Right Specs + Proper Back-Up

First, make sure that your computer runs under the right specifications, which are as follows:

Graphics have to be Direct-X 9 Capable Video Card with the WDMM Driver.

RAM has to be 2GB (64 Bit) or 1GB (32 Bit), and;

You need to have 20GB (64 Bit) or 16GB (32 Bit) of Disk Space;

Your Processor has to be 1GHZ CPU or higher;

Now, go ahead and back up the data that you have. . You can either save your data on external hard disk, or on cloud services such as *Dropbox, Google Drive,* or *OneDrive.*

For Windows8, you can simply back up personal files by clicking the *File History* button.

You can also create a System Image to back up everything on your computer, especially if you're using Windows 7 or 8. All you have to do is head to the Control Panel, and then choose *Back Up Your Computer*.

On the left side of the screen, choose *Create System Image,* and choose where you would like to save it.

Updates

Before installing Windows 10, you have to make an update first. To do this, just follow the instructions below:

If you're using Windows 7, go to *Start Menu* > *Control Panel* > *System and Security* > *Windows Update*.

For Windows 8, go to *Start* > *PC Settings* > *Update and Recovery* > *Windows Update.*

For other versions, just wait for the computer to send a prompt about the Update.

Wait for the Windows 10 Prompt.

Time to Install!

Take note that there are two ways to do this. Check out the first one below:

You can try doing this:

If your computer is running on Windows 7 or 8 or other earlier versions, you should first download a *Media Creation Tool* available from Microsoft. Check whether you need the 32-bit or 64-bit version.

Run the tool that you have downloaded, and then select *Create Installation Media for Another PC,* and then click *Next.*

Now, select *Language, Edition,* and *Architecture* before clicking *Next.*

When asked which media to use, choose *USB Flash Drive*.

Then, insert the flash drive into the USB Port, wait for a prompt, and click *Next.* The PC will now be downloading Windows 10.

After around 10 minutes or so, open the drive in the *Explorer* and then choose *Setup.*

Wait for the *Ready to Install* prompt, and then select *Choose What to Keep*.

Wait for the rest of the installation to finish and you’re all set!

Or, you can also try this one below:

The second way is perfect only for those whose computers are already running on Windows 10. This should be done if you’re not happy with your current setup, or you just want to completely refresh your computer. You can do it this way:

Click *Settings* from the Windows 10 Menu.

Click *Update and Security.*

Click *Recovery > Reset this PC > Get Started.*

Choose *Remove Everything.*

Wait for the rest of the installation to finish.

Chapter 2: Using the Start Menu

A lot of people find the Windows 10 Start Menu to be confusing—and this is really evident, especially in the beginning. But don't worry because in this chapter, you'll learn how to make the most out of that Start Menu—and make sure that it'll work for you!

You can use the Start Menu this way:

Click the *Start* Menu. This will appear on the left side of the screen.

Click *All Apps*. Again, this'll be on the left side of the screen. You'll then see a display of all the apps installed on your computer.

To search for an app or file, type what you're looking for in the *Search Field* and you'll see a list of choices popping up onscreen.

The *Power* button would allow you to rest or shut Windows down. This is found on the left column of the screen.

To lock the PC, just right-click your account name and then you'd see the following options: *lock, change account picture, sign out*. Choose lock.

You can also pin certain items on the Start Menu. To do this, just right click on the file you'd like to see on the Start Menu and then *click Pin to Start.*

To manage the tiles you see on the right side of the screen, just right-click on a tile, and then you'll see a menu pop up onscreen. Choose either *Unpin from Start, Resize* or *Pin to Taskbar*. Also check if there is an *Uninstall* option—this would come with most apps.

Using both Start Menu and Start Screen

If you’re quite a multitasker and you want to work on both the Start Screen and Start Menu at the same time, here’s what you have to do:

Click *Start* button, followed by *Settings* > *Personalization.*

Click *Start.*

Choose *Use Start FullScreen.*

Click Start Screen to make Start Menu disappear.

Uncheck *Use Start FullScreen* to return to *Settings.*

Then, you can also resize the Start Menu. To do this, click *Start.*

Move cursor to the top of the Start Menu, and then drag and move it up to the top of the said Menu. To decrease the height, just drag the cursor down.

Increase width by dragging cursor to the right, and drag it to the left to decrease.

Customizing the Start Menu

Take note that you can definitely customize what you see onscreen and how you want your PC to work for you!

To make a switch between the Start Menu and the Start Screen, open *Settings* > *Personalization* > *Start* > *Start Behaviors* > *Use Fullscreen Start when in Desktop.*

To change the color of the Start Menu, window borders, and taskbar, go to *Settings* > *Personalization* > *Colors*. If you want a brightly colored PC, just go to *Show Color on Start, Taskbar, and Action Center*, and it’ll happen.

To customize what you’ll see onscreen, go to Settings > Customize. There, you’ll see a list of suggested apps from Microsoft. Another setting would show you setting controls for your recently opened programs, and the last one would be about *Jump List* items.

Click *Start* > *Choose Folders* to choose which folders you'd like to see onscreen.

Click *Start* again to see whether you have all the folders and files that you need.

The Secret Menu

There is also a so-called "secret" Start Menu that you can also use and customize. Here's how you can gain access to it.

Right click on the *Start* icon.

You will then see a pop-up menu with mostly everything you can do with the computer!

If using touchscreen, you can access this menu by tapping and holding he start button for at least 5 to 10 seconds!

Chapter 3: Understanding Settings

By understanding the settings and knowing how to use it, you'll easily be able to navigate Windows 10 and make the most out of it!

Accounts

You can find the *Accounts* tab under the *Charms* bar, and is basically about your Windows account.

Your Account is your primary sign-in account. This is linked to Microsoft's Cloud Network.

Sign-in options will ask you how exactly you want to open your computer. You can make use of normal log-in plus password, choose Windows Hello, which would allow you to log in using biometrics.

Work Access tells you whether the PC or your account is connected to another network.

Family and Other Users allows you to add more admins to the PC. To do this, just click *Set up account for assigned access > Choose an Account > Choose an App.*

Sync Your Settings allows you to sync this PC with your other gadgets—and other computers at home, as well.

Privacy

Of course, you can also make sure that your privacy is protected with the help of Windows 10. Here's how:

General is about deciding whether you'd like your name to appear on apps,

programs, photos, and any other file that is connected to your computer.

Location basically works on GPS, and allows you whether you'd like to let others see where you are or not.

Microphone gives you the chance to turn the microphone on or off.

Speech, Inking, Typing mostly gives you the option to use Cortana (Windows 10's digital assistant) or not.

Account Info, Calendar, Contacts, Messaging, Radio just give you permission whether you'd like to sync them with other devices or not.

Other Devices gives you permission to sync Xbox One and information found there with your Microsoft account.

Feedback. Choose whether you'd want Microsoft to ask you for feedback *once a day, once a week, automatically, always,* or *never.*

Devices

The *Devices Tab* gives you options as to which devices you'd like to connect/sync with your PC.

Printers and Scanners allows you to add printers to your computer. To do this, just click *Devices and Printers> Device Manager> Related Settings> Add a Printer or Scanner* and follow the instructions you'll see onscreen.

Connected Devices, meanwhile, is about other connected devices that are not printers or scanners.

Mouse and Touchpad gives you a chance to configure Mouse and Touchpad settings. Just choose *Mouse and Touchpad > Related Settings> Additional Mouse Options.*

Typing allows you to choose whether you'd like to use a physical or onscreen keyboard.

Autoplay allows you to choose whether Autoplay should be switched on or off.

Ease of Access

The *Ease of Access Tab* contains Windows 10's Accessibility Settings.

Narrator helps you gain voice protocol. It basically reads what is shown onscreen, or whatever you type, and you can choose between *David* and *Zira* as your narrator. You can also add the sounds you want to hear, or the language you want to focus on.

Magnifier gives you a floating toolbar and also magnifies your screen.

High Contrast allows you to choose a High Contrast theme.

Closed Captions provides you with subtitles or captions to help you understand what you hear onscreen. It works for sites such as Hulu and Netflix.

Keyboard and Mouse contains well, Keyboard and Mouse settings—like what's in earlier versions of Windows.

Personalization

Then again, you also have the chance to personalize your Windows 10 Experience. Take note, though, that you'd have more chances to customize your experience—but more of that on a later chapter. Here are the basics first:

Background is mainly just the wallpaper of your computer, and choose how you'd want the photo to fit on your screen.

Colors are the colors that would be used for your desktop, toolbars, etc.

Lockscreen is what you'll see onscreen while it is locked. Click *Pictures*, and you'll see the 5 recent lockscreen pictures used, as well as a *Browse* button for you to choose photos from your files. Choosing *Slideshow* would make a slideshow of pictures as your lockscreen. You can also choose *Screen Timeout*, and more *Screen Saver Settings*, as well.

Themes would help you choose which theme you'd like to use. Go to *Classic Theme Settings* > *Related Settings* to do this.

Start helps you turn applications and notifications on or off—and more. You'd learn more about this in the next chapter.

Update and Restore

And of course, you also have the Update and Restore tab, which works something like this:

Activation is about the version of Windows that you have, and gives you the chance to change Product Key.

Advanced Options would give you more Update settings to choose from.

Backup gives you the chance to backup your settings.

Click *Check for Updates* so you could check for updates manually.

For Developers is all about making apps and programs while on Windows 10.

Recovery contains options that you can use to fix your computer, which are: *Reset PC, Go back to Earlier Build*, and *Advanced Startup*.

The tab named *Windows Update* contains everything you'd ever need to update Windows.

Windows Defender, meanwhile, is your cloud protection system.

Chapter 4: Understanding Interface

In this chapter, you'll learn more ways to enjoy Windows 10—and it's mostly about knowing how to work with its interface!

Using Quick Access

A lot of people say that *Quick Access* makes Windows 10 a whole lot more manageable—and there is a lot of truth to that. You can learn more about it below:

To add a file to *Quick Access*, just navigate towards the file you want to add, and then simply click Add to Quick Access.

To remove a file from *Quick Access*, go to the said file and click Unpin from Quick Access.

To remove recently used files and frequently used folders from *Quick Access,* just go to *View* > *Options* > *General* > *Privacy*. Then, uncheck the boxes that say *Show Recently Used Files*. Click *Clear* > *Clear File Explorer History*. You can also choose Hide or Hide from Recent.

To change the way File Explorer opens, just *click View* > *Options* > *Open File Explorer* > *This PC.*

Using Snap Assist

This is a feature that is exclusive for Windows 10! This helps you snap a certain window to a certain side of the screen so you won't spend lots of time moving it around.

To snap a window with the mouse, click on its title and drag it towards the side of the screen. You will then see an outline that will show you where the window would

appear once you have dragged it.

To snap with the keyboard, just press Windows Key + Left Arrow (or Right Arrow).

To snap to one of the quadrants, just press Windows Key + Up Arrow (or Down Arrow), and then move it around by pressing Windows Key and arrow keys together.

Using Multiple Desktops

Yes, you can make use of multiple “desktops” while using Windows 10. To make this happen, just follow the instructions below:

Add a desktop by clicking Task View. Press Tab + Windows Key > New Desktop.

Now, you have two virtual desktops. To switch between them, just press Windows Key + CTRL + Left Arrow + Windows Key + CTRL + Right Arrow.

To move windows between desktops, just right click on the window you’d want to move, then choose where you’d want to move it to.

To close the desktop, just click X or press Windows Key + CTRL + F4.

Hiding System Tray Icon

To hide the *System Tray Icon* temporarily, all you have to do is drag and drop said icon to the hidden area found under the arrow. However, if you can see all of the icons onscreen, just open *Task Manager* (press CTRL + ALT + DELETE) and then go ahead and terminate GWXUX or GWX.

To hide the *System Tray* permanently, go *to Control Panel > Windows Update.*

Choose *Installed Updates*—you’ll see this on the left side of the screen.

Look for the label *Update for Microsoft Windows KB3035583* and remove said update.

Always skip this update by hiding it from future updates.

Part 2: Second Week

Now that you know the basics, it’s time to enjoy Windows 10 even more!

Chapter 5: Understanding the Features

And of course, in order to make sure that you enjoy the various features of Windows 10, it’s just right that you learn how to use its many features!

Writing on Web Pages

This is a fun feature of Windows 10! Basically, you can have a little fun by literally writing on web pages, with the help of Windows 10’s *Microsoft Edge*, a next-gen browser!

To start “inking”, click on the icon that looks like pen and paper found on the upper right corner of the browser window.

Once you have clicked that, the web page you’re browsing will refresh. You can then start using the tools you see onscreen to write or doodle on the said webpage!

To type comments, you can use the text tool found on the toolbar mentioned earlier.

You can declutter the page by clicking on the numbered pin and closing the text box.

Click the trash icon to delete the text box, as well as the comment pin.

You can also use the clipping tool to make crosshair out of your cursor.

You can also erase line by line using the eraser tool. Just click, hold, and move over the lines you don’t want to see onscreen.

To share or save what you have done, just click the save icon. Then, a pop-up menu will appear onscreen. This’ll ask you whether you want to add the project to your

reading list, save to *OneNote*, or add to your favorites. Just choose what you want to happen.

Managing Notifications

Since Windows 10 proves to be the Operating System for the new age, you can expect it to give you real time notifications. In short, you'll get notifications from *Facebook, Twitter, Instagram,* and any other apps you might be using—as long as they're connected to your Microsoft account.

To choose which notifications you'd like to have, go to *Action Center* > *Show Notifications from These Apps*. There, you'd see a list of the apps you have. Just choose what notifications you'd like to have and you're all set!

You can also choose which Quick Actions you'd like to have access to. To do this, go to *Settings* > *System* > *Notifications and Actions* > *Choose Your Quick Actions.*

Sharing Files Quickly

Look for the file that you want to share.

Click *Sharing* in the *File Explorer.*

Click *Share* button.

Choose the program you'd want to share the said file with.

Configure options by going to *Settings* > *System* > *Share Lab.*

Setting Default Programs

You can change default programs and protocols by following the instructions below:

Open *Settings* > *System* > *Default Apps*.

Change the programs you want to use for email, calendar, maps, web browser, video player, photo viewer, and the like.

To set individual file types, go to *Settings > System > Choose Default Apps by File Type.*

To set defaults for protocols, go to *Settings > System > Choose Default Apps by Protocol.*

To change default programs, just go to *Settings > System > Set Defaults by App > Set Default Programs > Set Program as Default > Choose Defaults for this Program*

Exploring Photos App

Take note that the Photos App now has two main features, and these are: *Collections* and *Albums.* Your photos are chronologically arranged by date in the *Collections* tab. Meanwhile, *Albums* contain albums that the app has created automatically for you.

To add a folder, go to *Settings > Sources > Add a Folder > Add this Folder to Pictures.*

To show photos and videos from *OneDrive,* just choose Settings > Show my Photos and Videos from OneDrive.

To share pictures, just select the picture you want to share and click *Share*!

You can also make use of Filters and other Editing Features, too!

Delaying Shutter Speed with Pop-Ups

You can also delay shutter speed in capturing screenshots by making sure that you get to capture pop-ups, too. Here's how:

Open *Snipping Tool* and then click *Delay*. Choose between numbers 0 to 5.

Choose the type of *Snip* that you'd like to make by clicking what you find next to

New. Choose from window, rectangular, free-form, full-screen.

Click *New* so you could begin snipping. You will now have an allowance of 0 to 5 seconds, depending on what you chose earlier. The screen will freeze and you'll be able to capture the image you want.

Click *Save* to save your screenshot.

Importing Bookmarks

If you have been using other browsers before and want to regain access to bookmarks you've made there, you can just import them to the Edge. Here's how:

Open Edge and click "…", then click Settings.

Choose Import Favorites from Another Browser.

Choose all the browsers you want to import bookmarks from and you're all set!

Sideloading Apps

Sideloading is important because it allows you to install apps that are not available in the Windows Store. Here's how you can do it:

Open *Settings > Update and Security.*

Go to *For Developers > Sideload Apps.*

You will now receive a warning about sideloading being dangerous. Just click *Yes* to say that you understand the risks.

Disabling Wi-Fi Password Sharing

In order to make sure that your Wi-Fi connection stays yours, and that you would have more privacy, you can disable Wi-Fi sharing! Here's how:

Go to *Settings > Network and Internet > Wi-Fi > Manage Wi-Fi Settings.* You can also turn off networks that automatically connect to Skype, Quora, Outlook or

Facebook, as well.

Booting in Safe Mode

In case you are experiencing problems with Windows 10 and have to revert to Safe Mode, here’s what you have to do:

Open *Settings > Update and Security > Recovery > Advanced Startup.*

Click *Troubleshoot > Advanced Options > System Restore.*

Click *Restart,* and press *F4.*

Chapter 6: Keyboard Shortcuts and More Tricks

Learn keyboard shortcuts and tricks that will help you use Windows 10 in a better, more manageable manner!

Shortcuts for Opening Programs

Go to *Start Menu* > *All Apps.*

Look for the app you want to make a shortcut for and then right click on it. You will then see a dropdown menu. Choose *Open File Location*, and then skip the next step that will pop up onscreen.

Once you've found the app, click and drag it from the *Start Menu* all the way to the desktop. Right-click and then choose *Properties.*

Now, the *Properties* window will open up onscreen. Look for the Shortcut tab and then choose Shortcut Key. Tap the key you'd want to be associated with the app (i.e., CTRL + ALT + [chosen key]).

Click *Continue*.

You can now use your chosen shortcut to open this certain app!

Command Prompt Shortcuts

Go to Start Menu > All Apps > Windows System > Command Prompt.

Click Properties > Options > Edit Options > Enable CTRL Key Shortcuts.

Now, here's a list of shortcuts you could use:

Alt + F4 (close command prompt)

CTRL + A (select all in the current line)

CTRL + C or CTRL + Insert (copy selected text to clipboard)

CTRL + F (open Find Window from the Command Prompt)

CTRL + M (enter mark mode)

CTRL + Page Up/Down (move screen one page up or down)

CTRL + Shift + Home/End (move cursor to the beginning/end of screen buffer, and then select text and beginning/end of output)

CTRL + Shift + Left/Right (move cursor left/right and select text)

CTRL + Up/Down (move one line up or down)

CTRL + V or Shift + Insert (paste copied text)

Shift + Home/End (move cursor to beginning/end of current line and select text)

Shift + Left/Right (move cursor left/right one character and select text)

*Shift + Page Up/Down (*move cursor up/down screen and select text)

Shift + Up/Down (Move cursor up or down one line and then select text)

Up/Down/Left/Right (In mark mode; move cursor up, down, left, or right)

Other helpful Keyboard Shortcuts:

Windows Key + Left (Snap Window to Left Side of Screen)

Windows Key + Right (Snap Window to Right Side of Screen)

Windows Key + Up/Down (Snap Window to Quadrant)

Windows Key + Tab (Task View)

Windows Key + CTRL + Left (Go back to previous virtual desktop)

Windows Key + CTRL + Right (Go to next virtual desktop)

Windows Key + CTRL + F4 (Close current virtual desktop)

Windows Key + CTRL + D (Create new virtual desktop)

Speeding Up Windows 10

To create faster setup, just go to the *Search Field*, type *netplwiz* and press *Enter.*

Once you see the label *Users must enter a username and password to use this computer*, look for the box next to it, and unclick it.

To get a faster *Start Menu*, go to the *Search Field*, type *sysdm.cpl*, and press *Enter.*

Click *Advanced > Settings.*

Uncheck *Activate Windows when Minimizing and Maximizing.*

Click *Apply.*

Now, to shut the computer down faster, just right click on the desktop, click *New > Shortcut*, and then type the following in the location field: *%windir%\system32\shutdown.exe/s/t/0*

Maximizing Windows 10 Use

Make sure to minimize all windows except for the one that you're using. This will speed up the process and make sure the computer would not lag!

Pin the Recycle Bin to the Start Menu—just like the old times!

Use the Alt Key to quickly open the Properties Menu.

Speed through the Start Menu by clicking All Apps and then clicking on the numbers and letters you see.

To scroll on the background, just click *Settings > Devices > Mouse and Touchpad*

> *Scroll Inactive Windows* when I hover at them.

Chapter 7: FAQs

Finally, here are frequently asked questions about Windows 10—answered for your convenience!

Q: When was Windows 10 Launched?

A: Windows 10 was officially launched in July 28, 2015. This makes it FREE for one year for those whose operating systems are Windows 7 or 8. If after a year you have not upgraded to Windows 10 and decide that you want to, you'd have to pay $119 for Windows 10, and $199 for Windows 10 Pro.

Q: Where can you get Windows 10 Apps?

A: Microsoft believes that you can buy from the Windows Store once and then these universal apps would work on all of the gadgets that you have synched with your account. Even Microsoft Office will work—so you'd basically be able to live a life of convenience!

Q: Can it work with laptops or Windows Phones?

A: Windows 10 can be the sole OS for your desktop/laptop computer, Windows Phone, and Tablet. This is because of *Continuum*, the said driving force that makes Windows 10 work—and which makes it easy to operate whether by using mouse, keyboard, or touchscreen. There are finger-friendly fullscreen apps that you could download and use, too!

Q: What if I don't want the apps to take all of the screen space, or what if I want to re-arrange them?

A: The great thing about Windows 10 is that fullscreen apps are optional—so if you

feel like your device is not compatible with them, you can opt not to download them. Since we now have *Continuum*, you can expect that the apps will work on the gadget you want it to work on—and that it will take its shape so you would not have a hard time to adjust.

The Start Menu is also reminiscent of the good old days and is not as confusing as the one from Windows 8 (which a lot of people hated). Plus, as you may have learned earlier, this new start menu makes use of Live Tiles that you can customize for your own benefit and preferences—making Windows 10 the best operating system you could have.

You can also arrange apps side by side, thanks to what is called *Snap.* This is what the quadrants are for, and as you can see, you've been given commands that you can use to make it work. The great thing is that this supports multiple virtual desktops, so if you multitask a lot, or have lots of things to address, this will certainly be helpful for you!

Q: Do I have to use Windows Explorer? Or can I use other web browsers instead?

A: Internet Explorer, while still part of the Microsoft Brand, is no longer the main browser used—so you can breathe a sigh of relief, especially if you've found it to be extremely slow in the past. Now, you'll be able to use Microsoft Edge (as you have been told earlier), which has lots of new functions and extensions that will deliver a great experience for you.

Microsoft Edge also comes with *Cortana*, your very own virtual assistant that will provide suggestions about what to put in the address bar, or gather information about the restaurant you want to visit, or tell you about latest promos and the like. This way, you'll always be informed—and will never get bored anymore!

Q: Is this really better than Vista? Or should I just stick to Windows 7 or 8 instead?

A: Well, Windows 10 is definitely better than Vista as it does not lag like that and works on more computers, and also has a lot of upgrades from Windows 7 and 8. Some users have said their laptops' batteries have been draining ever since they used Windows 10, but noticed that when they charge their laptops better, then it does not happen.

In short, it would still be best to make an upgrade because sooner or later, apps in Windows 7 or 8 might be phased out, and Windows 10 really works faster, and works in a sleek, professional manner, too!

Wordpress Bootcamp

Learn the Basics of Wordpress in Two Weeks

Table of Contents

This document is geared towards providing exact and reliable information in regards to the topic and issue covered. The publication is sold with the idea that the publisher is not required to render accounting, officially permitted, or otherwise, qualified services. If advice is necessary, legal or professional, a practiced individual in the profession should be ordered.

- From a Declaration of Principles which was accepted and approved equally by a Committee of the American Bar Association and a Committee of Publishers and Associations.

The information provided herein is stated to be truthful and consistent, in that any liability, in terms of inattention or otherwise, by any usage or abuse of any policies, processes, or directions contained within is the solitary and utter responsibility of the recipient reader. Under no circumstances will any legal responsibility or blame be held against the publisher for any reparation, damages, or monetary loss due to the information herein, either directly or indirectly.

The information herein is offered for informational purposes solely, and is universal as so. The presentation of the information is without contract or any type of guarantee assurance.

Introduction

I want to thank you and congratulate you for downloading the book, *“ Learn the Basics of WordPress in Two Weeks ”* .

This book contains proven steps and strategies on how to start using WordPress to build your blog or website.

WordPress is the most popular website platform today, mainly because it is easy to use. Even without prior knowledge in web design and development, you can teach yourself in using WordPress.

In fact, this book aims to teach you to start using WordPress in just two weeks or less.

In this book you will learn:

What Wordpress is

How to use Wordpress

Common Wordpress features

And much more!

Thanks again for downloading this book, I hope you enjoy it!

Click Here to get step by step instructions for setting up your Wordpress site with Bluehost.

Before we start…Bonus: Receive More Free Books By Me!

Thanks for reading my book! I hope you received awesome value from it. I'd like to thank you by giving you more free books, and info related to programming and technology!

Simply click on the link below and enter your email address on my website!

Everything you will get will be useful to you if you're serious about changing your programming skills. I will be offering:

More Books By Me

How I was able to learn programming in record time

Info On My Favourite Products Related To Programming

And Much More!

Click Here to get access to all this info!

Chapter 1 – Why Learn WordPress?

Many people think that WordPress is just another blogging platform. This misconception is due to the fact that in the past, WordPress is really for blogging. But through the years, WordPress has evolved as a dynamic content management system (CMS). Although you can still use this CMS to create a basic blog, it now allows you to build a fully functional website.

The best thing about WordPress is that it is quite flexible and easy to use. This is the primary reason why WordPress is the most popular CMS today. About 23% of websites that are active today on the internet are powered by WordPress.

Because of its versatile features, popular brands are using WordPress to power their websites such as eBay, The New York Times, Disney, Facebook, Time Magazine, CNN, LinkedIn, Sony, and Google.

Let’s take a glimpse at why you should learn and use WordPress

WordPress Gives You Freedom

Basically, WordPress is a free software. You have the freedom to download, install, modify, and use it any way you can. You can build any type of website. It is also an open source platform, which means that the source code is readily accessible for anyone to study, modify, and experiment with.

Millions of people around the world are using WordPress, and every day, new people are creating websites powered by this outstanding CMS. The main reason why people can easily adapt to WordPress is because it is very easy to use and learn.

You can Easily Manage WordPress

WordPress features a built-in updater, which makes it easy to update your plugins and templates through the admin dashboard within the WordPress. It will also notify you if there's an updated version of WordPress, which you can easily update through several mouse clicks. All content are safe if you set up regular backup.

You can Extend WordPress through Themes and Plugins

It is interesting to take note that most people who are using WordPress are neither programmers nor web designers. In fact, thousands of individuals have started using WordPress without any fundamental knowledge of website design.

The main reason why WordPress is such a dynamic platform is you can choose from literally thousands of free themes (web templates) to improve the look of your website. Hence, you can give your website its design and feel according to its purpose. You can find any WordPress theme for just about anything. You can learn more about this in Chapter 2.

WordPress also offers superb flexibility and could be extended through plugins. Similar to themes, you can use thousands of plugins to enhance your website. These plugins can enhance the functionality of the WordPress and could add a whole new platform to your website. You can learn more about plugins in Chapter 2.

WordPress is SEO Friendly

WordPress has been developed using high quality and standard compliance code that produces semantic markup that makes your site search engine friendly. WordPress design and themes are very search engine friendly, and it is possible to make it even more attractive for SEO by using SEO plugins.

WordPress Could Integrate Various Types of Media

WordPress offers a dynamic platform, which allows you to use different media types. It

features a built-in support that allows you to handle video, audio, and photo content. It also supports embedding features that allows you to embed videos from external sites such as YouTube and Vimeo.

WordPress Offers Reliable Security Features

Web security is one of the fundamental elements of WordPress, so you can securely run any website through this platform. But with the vastness of the World Wide Web, it cannot be certain. Internet intruders are out there who want to get inside as many websites as they could. The good thing is, WordPress is open for security plugins to avert any form of security threats.

The Different Ways to Use WordPress

You can use WordPress in numerous ways. Many business websites are running on WordPress and serves as a crucial aspect of their sales generation. You can use WordPress as the following:

Blog (Personal or Business)

Online Arcade

CMS

Portfolio

Gallery

Online Store

Membership portal

Video curation site

Rating site

And many more!

I hope this Chapter answered your question about why you must use WordPress. The best way to learn WordPress is start using it. In the succeeding Chapters, you can learn basic elements of WordPress every day.

Chapter 2 – First Week: Setup, Dashboard, Admin Bar, Settings, Links and Images

WordPress is easy to learn as long as you are can spare one to two hours every day for two weeks. You only need very basic internet skills and of course the capacity to follow instructions accurately. You don't even need to learn HTML or other programming languages to create your website. Even though programming and coding skills will come in handy in the future, these are not necessary to start using WordPress.

Millions of individuals around the world started using WordPress from scratch. In fact, most WordPress users don't even know how to design websites or write a code.

The mere fact that you have reached this Chapter is a good sign that you are willing to learn WordPress. So, let's start.

Week 1 Day 1 - Learn Initial Setup

The first step is to learn how to install and setup WordPress. You need to choose between WordPress.org and WordPress.com.

If you want to build a personal blog and you don't have any intention to make money from your websites, then it is ideal to choose WordPress.com. But if you want to be a professional blogger, then it is better to use the self-hosted WordPress.org. The latter option will cost you money, but it is actually cost-effective for creating your own site.

For example, let's assume that you have chosen WordPress.com, and you purchased a custom domain ($17 annually), opted for a custom-design ($30 annually), and paid to remove ads ($29.97 annually). The total cost of $76.97 per year will not provide you full control.

If you have chosen WordPress.org, you can choose to buy hosting from Bluehost

(endorsed by WordPress) that will cost $47.4 annually, and it already includes a free domain name. If you choose other web hosting companies, you may need to pay the same price, but you can't obtain a domain. You can obtain a domain name for only $10 from NameCheap or GoDaddy. The total cost of $57.4 annually will provide you full control.

Click Here to get step by step instructions for setting up your Wordpress site with Bluehost.

You might be a bit concerned if the one is easier to use than the other, don't worry, their interfaces are similar. It is ideal to use WordPress.org so you can have full control of your website. But still, you need to choose the WordPress platform that is suitable for your preferences and needs.

After choosing the platform, the next step is to select your domain and hosting. Take note that your domain is your web address that a user need to enter into the search bar to visit your page. Then, you need to select web-hosting service, so other internet users can access your site.

After signing up with a web-hosting provider, you can install WordPress using SimpleScripts, especially if you have chosen BlueHost.

Week 1 Day 2 - Dashboard Basics

After installing WordPress, you can start building your custom website or blog. However, before posting any content. It is best to familiarize yourself with the dashboard basics. The dashboard is located on the left side of the screen, which you can easily see after you log in to your wp-admin account. Most users refer to this as the back-end of your website.

Using the Dashboard's Home Page, you can easily access the content of your website and take a look into other areas of the WordPress platform. The Dashboard presents information in separate blocks called Widgets. The default WordPress widgets are the following: Welcome at a Glance, Quick Draft, Activity, and WordPress News.

The Welcome widget shows the links form the top tasks you can perform in creating a new website.

The At a Glance widget offers a short glimpse of the number of pages, posts, and comments on your website. Every content types are shown in link form, and once you click them you can be directed to the particular area so you can easily manage the content.

The QuickDraft widget allows you to write a new text content easily. You can quickly write a title, add media, enter the content, insert tags, save your draft, or Publish right away.

The Activity widget shows the most recent comments, published posts, and scheduled posts.

The WordPress News widget features the latest news, articles, and announcements from the official blog of WordPress.

You can expand, collapse, and rearrange widgets by hovering the mouse cursor on the title bar. If the mouse transforms into a 4-arrow icon, you can hold the left-mouse button and drag the widget to the area you want to move it. This is known as a drag and drop functionality.

You can also add new widgets through plugins or through a special program code.

Week 1 Day 3 - Admin Bar

The Admin Bar is located at the topmost part of the screen. The WordPress logo is located on the far left of the Dashboard. Hover over the logo, and you will see the links to specific information about WordPress. These links can help you if you need to visit the support forums or provide feedback.

The name of your site is located to the right of the logo. You can hover on this and you will see a submenu to access the front end of your website. The next area provides a quick

glimpse for comments.

On the left side, you can find the link for + New. Once you hover over this link, a submenu will appear that will redirect you to the Add New screen for user, page, media, and post. You can also do these things using the navigation on the left area of the Dashboard, but if you want to do things quickly, these links could help you a lot.

You can also see the section displaying the text “Howdy, your name” on the far right area. You can hover over this area to access the submenu, which you can use to log out from your site or edit your profile.

Front End View of the Admin Bar

If you are currently logged into your WordPress website, the Admin Bar is visible on your website’s front-end site. Try hovering over the name of the site, and click visit site, and it will redirect you to your website’s front end.

Once you hover the title of the site section, you can see the links for the dashboard that will take you back to the dashboard home, as well as the links to take you to the header, background, menu, widgets, and themes. Take note that these sections can also be found on the navigation pane on the left area in the dashboard.

How to Turn Off the Admin Bar View

The Admin Toolbar is generally useful, but there are times that you may want to turn it off to see the full view of your website’s front end. Take note that the Admin Bar will only appear if you are logged in to your website as admin.

If you want to turn it off, just click the Edit my Profile function. A screen will appear where you can find a checkbox next to the text: Show Toolbar when viewing site . All you need to do is to uncheck the box, and scroll down to the bottom and click Update profile to save the changes.

Week 1 Day 4 and 5 – Explore WordPress Settings

On the Dashboard, find the Settings menu and hover on top. A sub menu will appear where you can access several options for WordPress settings: General, Reading, Writing, Media, Discussion, and Permalinks.

General Settings

To explore around, you can go first to the General Settings. The first elements that you will see in this option is the Site Title and the Tagline. Make certain that these texts match to your website, because your site title will be used for SEO. As a default, WordPress displays “just another WordPress site” as the tagline. You need to change this tagline to describe your website, as it is important for the SEO of your website.

Next is the WordPress Address (URL) . For the website address URL, you can enter the URL address if you want your website homepage separate from the directory where you have WP installed. At this point, it is best not to touch these URLs.

Then, you will also see the Email Address , which you will use for admin purposes, including new user notification. There is also the Membership settings. WordPress allows online users to sign up to your website. This is a useful feature, if membership is crucial for your website.

Meanwhile, the New User Default Role is set to subscriber. Don’t touch this setting for now, because you might accidentally grant admin privileges to anyone who sign up for your website.

Then, there is the Timezone. Browse through the list to choose the city within the same timezone in choosing your preferred date format. Take note that this format will appear on your posts. After you have updated these settings, don’t forget to Save changes .

Reading Settings

The Reading Settings show the settings, which will affect the content display. Here you can tweak the display of your website's front page – either a static page or your latest blog post. After building several pages, you can list these pages in this setting so they will appear on the front end.

The next setting section will allow you to control the content display in RSS feeds, which includes the max posts to display and if you want only to show a summary or full text. Search engine visibility is the last section. If you want search engines to not index your website, you should check box to Discourage search engines from indexing this site. This is a great setting if you are still on the development phase indexing is still not recommended.

Remember to Save Changes .

Writing Settings

The Writing Settings affect the writing and publishing for your website. The section on the top allows you to control the editor inside the Dashboard, while the other sections allow you to control external publishing methods.

The first section shows the options for default categories formatting and posts format. You will also see the Press This bookmarklet section, which allows you to easily blog about the things you find interesting on the Internet. If you want to use this, you first need to drag the link on the screen to your browser's bookmark bar. If you are on another site, just click on the bookmarklet to access a popup window to share the content through your blog.

Meanwhile, the Post via email settings will let you send an email to your website with post content. In order to use this setting, you should set up a secret e-mail account using a POP3 access. The last section can be used to update services. Once you publish a new post, WordPress will instantly notify you of the updates. Be sure to Save Changes .

Media Settings

The Media Settings page allows you to set max sizes for images you want to add into the post. These settings will allow you to save time if you want images to be of the same size, if you want to set default settings for image sizes.

The option for Uploading files allows you to choose if you want your uploads to be organized into month or year. Make certain to Save Changes .

Discussion Settings

The Discussion Settings allow you to manage the comments and links to your posts and pages.

The first section is the default article settings . The first setting allows you to manage links you make to other blogs. The second setting allows you to manage the trackbacks and ping backs or links back to your site. Finally, the third setting allows your site visitors to post comments on new articles. You can uncheck this box if you don’t want your visitors to comment on the posts.

The Other comment settings allows you to choose the guidelines for how visitors post comments and how you manage their comments.

For the email me whenever section, you can select to be notified via email if someone posts a comment or if a comment has been handled by a moderator.

The section for Before a comment appears will allow you to manage the published comments. Through this, you can select if an administrator should approve the comments first or if the comments could be published automatically.

In the section for Comment Moderation, you can change how a comment is managed depending on the number of links. Inside the box, you can include IPs, emails, names, words, and URLs to filter comments for moderation. This feature alongside the comment

blacklist are can help you in defending your blog against comments that are inappropriate and spammy.

There is also the Avatar setting. You can enable the avatars of users who posted comments on your website, select a default avatar for users who don’t have their own avatar yet, or filter their comments based on their ratings.

Again, be sure to Save changes .

Permalink Settings

Permalinks refer to the permanent URLs to each age and blog post including tag and category archives. In general, a permalink is the web address that can be used to link to your content. It got its name by combining permanent and link. Hence, permalink.

The Permalink settings in WordPress allows you to manage the default structure of your permalinks. You can either select from common settings or build your customized URL. As a default, WordPress utilizes URLs that have question marks and with numbers. You may want to change the permalinks here to improve its forward compatibility, usability, and aesthetics.

Week 1 Day 6 –Create a WordPress Post

To create your first post, find the Posts menu in the left side of the Dashboard. You can click to expand it to access the submenu or just hover over the link Posts.

In the + New link in the Admin Bar, you can also find the Add New Posts page . Just click the link for the Add New to access the Add New Post page so you can start creating your first WordPress post.

You can enter the title of your post in the first box.

Below that is the post editor or Post formatting section , where you will type or paste the post content. There are two tabs on the right side of the tabs showing two modes of post

editing: Text and Visual .

The Text tab will let you access a plain-text HTML version of the post editor. You can use this mode to edit the HTML code of your post. If you are not comfortable with HTML, the visual mode is recommended.

The Visual tab is ideal for developers who are comfortable with the WYSIWYG (What You See Is What You Get) concept. It comes with a format toolbar with different options to format your posts. Many of these icons will look familiar if you have been using Microsoft Word.

The Publish box is located at the top of the right column of the screen. You can save your post as a draft if you want to save it. The Preview button will allow you to take a glimpse of how the post will actually look once you publish it. The Status will let you know if the post is currently a draft, has been saved, already published or if it is a pending review if you have schedule the post.

The succeeding links allows you to see the post's visibility or what will the site visitors see. The Publish line shows if the post will be immediately published or at a future date. The next section are for tags and categories assigned to your post. Save this lesson later.

If you want to change the screen options for your editor, you can click the screen options tab found in the upper right. This will expand the link and you will see the options, which will be displayed on the screen's post editor.

In the post editor, you can just drag or drop the arrangement of these boxes, to change how you want them to appear on the page.

If you need help in adding new posts, you can just click the Help tab found in the upper right. From here, you can easily refer on inserting media, using the editor, helpful tips for adding a post title, and pointers on customizing your post display.

Week 1 Day 7 – Adding Links and Images

Links are important for the SEO of your content, while images are ideal to improve the appeal and readability of your posts. WordPress makes it easy to add these two elements into your website.

Adding Links

In the page editor or WordPress post, choose and highlight the text, which you want to be added with link. Then, click the hyperlink button located in the toolbar. This will reveal a box that you can use to enter the specific URL of the hyperlink as well as the title which will be shown if you hover over the link. Clicking this checkbox will also allow you to access the link in the new tab.

If you prefer linking to an existing content on your site, you can just click to expand this section. With this, you can see a search bar to search your website, as well as a list of present pages or posts.

Choosing one of the items on the list will also change the hyperlink. If your hyperlink is ready, just click the Add Link button. Your chosen text will now have a link. If you want to remove the link, choose the text again and choose the unlink button. You will see that the link has been removed and the text is in normal display again.

Adding Images

Choose the best place in the post to insert your image. On the upper section of the Post editor toolbar, you will see the button Add Media . Click this button to reveal the Insert Media box.

WordPress stores a media library to keep all your media files such as videos and images. In this box, you can select to either upload a new file or use a present file from the media library. If you want to upload an image, click the button Select files and find the image

you want to use.

After your image has been uploaded, you will notice that it has been added to the media library. On the right area of the box, you will see the attachment details for the page. In this section, you can see the Description , Alt text , Caption and Title . You can also select the image alignment such as right , left , or center .

Ensure that the image has a checkbox. Click the insert into post button . Now you can see that your image has been inserted into your post.

Once you click the image, you will see two boxes in the upper section of the image. The first is the image icon. When you click this icon, you will see that another box will appear where you can edit all the image details. In this prompt, you can modify the image size based on the percentages or you can change the link URL, caption, alt text, and title. You can also choose the image link or to link the actual file of the image.

In choosing the advanced settings tab, you will see the link of where the image is currently hosted, properties and style that adds padding on the image, the CSS Class, and the actual height and width of the image in pixels.

Next, return to the image. The next icon in the upper corner will let you delete the image from the post. In clicking the Add Media button , you will notice the same insert media box. On the left side, you will see the options to set up a gallery and choose the featured image. If you have enough images in the media library, you can add a whole image gallery into your post.

Featured images are used by some themes. If your chosen theme has the featured image, go ahead and select the image and go back to the post editor. Click the preview button and check if the image has been inserted.

Chapter 3 – Second Week

Week 2 Day 1 – Categories & Tags

You can organize your posts into various categories such as topics or subject areas. For instance, if you are writing a post, consider a wider subject of the post. If you are planning to write more posts about the subject in the future, it is ideal to organize these posts together.

Adding New Categories

If you are yet to include categories, click the link showing + Add new category . Insert a new category by clicking the Add new category . You will notice that a new category has been included to the list of categories. To include a category to a post, click the checkbox next to the category.

Another method to manage post categories is by navigating through the link Posts > Categories that is located in the Dashboard menu. If you click the Categories link, you can access the Categories page. From this, you can view all the categories located on the right, where you can also include new entries.

Category Description

The description area allows you to add more information about the posts linked to that category. Some WordPress themes display this information.

Category Hierarchies

Unlike tags, categories could be nested into hierarchies. For instance, you may have a News category, and under that you may include sub-categories for World News and Sports News.

Category Slugs

The Category Slugs is the link-friendly version of the category. This is often in lowercase form and involves only numbers, hyphens, and letters.

What's the Difference between Categories and Tags

Just think of your website as a book. Categories are similar to the Table of Contents, while the tags serve like the index section.

Week 2 Day 2 – Create a WordPress Page

The first step in creating a new page is to locate the Pages menu in the Dashboard. Just click the Add new option.

You will notice that the page editor has the same interface with the post editor. The only difference lies on few boxes that you can find on the right section of the screen. Add the page title such as Contact Us. Remember, if you have created pretty permalinks, the page title will also be used as the URL slug.

After adding content for the page, you can Publish the section of the page editor, which is similar when you are writing posts. If you are ready to publish, you could either publish immediately, schedule for future posting or schedule it.

The section for Page Attributes includes a parent page and template to your new page. In the Parent section, you could organize the pages according to the hierarchies. For instance, you can create this new page with added pages below it. Take note that there are no restrictions on the number of levels that you want to nest pages.

The Template section allows you to include a template for your new page. This is possible if your theme allows custom page templates.

The Order interface lets you organize the page in numerical order. Pages are often organized alphabetically, but you can select your own arrangement by providing a number in this box.

Preview the page first, then hit Publish if all is well.

Week 2 Day 3 – Applying a WordPress Template

There are WordPress themes that feature templates, which change the format of the page on the front end of the website. Through page templates, the theme can provide you some flexibility for how the page will look and where specific elements will be placed.

The first step in adding a page template is to browse the Pages menu in the Dashboard and find the edit or add a new page link. Locate the Page Attributes area, where you can see a drop-down list for templates that you can use. Depending on the WordPress theme, you can see certain options for page template located in the drop-down menu.

To see how a template will change the look of the page, choose a template and preview it to check if it is appropriate for the page.

You can hit Publish once you have finally selected the template you want to use.

Week 2 Day 4 – Installing a Plugin

Plugins allow you to add more functionalities for your WordPress website. Installing plugins is easy.

The first step is to find and expand the Plugins menu after logging into your site Dashboard. Look for the Installed Plugins page where you can see a list of the plugins that are already installed on your site. Some plugins are automatically installed when you install WordPress such as JetPack and Akismet.

To add a new plugin, just click the Add New option. From this page, you can choose to browse for plugins that you want to install from the WP Plugin Directory. You can also filter your search according to the plug-ins published date, popularity or those that you have marked as your favorites.

To start uploading a plugin, follow this command thread: Choose file > locate your plugin

zip file > Install Now > Activate.

The last menu section under the Plugins in the Dashboard is the Editor , which you can use to change some elements of the plugins by modifying separate PHP files. But remember, if you make some modifications, the plugin updates will override your changes. Don’t change the plugin PHP if you don’t know PHP code.

Week 2 Day 5 – Managing WordPress Themes

The first step in managing WordPress themes is to browse the Appearance menu in the Dashboard. Choose the page for Manage Themes .

At the upper part, you can see the Active theme as well as the Customize option. When you click this link, you can see a preview window, which allows you to make some changes in the Tagline and the Site Title.

If you go back to the Manage themes tab, you can view a list of available themes under the active theme. These themes are already installed on your WP site.

If you choose the Live Preview link beneath any theme, you can see a preview of how your website will look with that theme. In the preview, you can browse different pages to see how the theme could handle templates, archives, and posts. On the left side of the preview window, you can edit the settings of the theme. These settings will vary according to the type of features included in the theme.

Be sure to hit the Save & Activate button found on the upper left-side to accept the new theme.

Week 2 Day 6 – Create a Custom Menu

You can create a custom menu that will serve as a navigation menu for your website. The Menu feature allows you to build your own custom menu as a replacement for the default menus of your chosen theme.

Some custom menus contain links, categories, pages, and other types of content. You can also include a custom navigation label for a menu item as well as include other attributes.

In general, there's no cap on how many menus you want to set up. Hence, if your them includes more than one menu location, you can select which custom menu to link with every location. You can also utilize custom menus alongside the Custom Menus widget.

How to Create a Custom Menu in WP Website

The first step is to expand the Appearance menu on the left side of the Dashboard. Hit on the link for Menus in the Appearance, which will activate the Menus editor page. Provide the name that you want for the menu, and click Create menu. Then, you can add menu items inside the boxes on the left such as links, categories, and pages.

If you want to change which menu options you see from this page, you just need to expand the Screen Options tab. Then, add other menu options such as CSS classes, formats, tags, and posts.

Once you have organized your menu items, make sure to Save Menu

Week 2 Day 7 – Managing Comments

Managing comments in a site built on WordPress is similar to the manner of handling pages and posts.

In the Dashboard, look for the Comments page. You can customize this screen in similar way as you customize other WordPress screens. A yellow color indicator signifies that the comment needs to be moderated. You can respond on comments through the Bulk Actions or by hovering on the action links.

In the Author column , aside from the name of the author, blog URL, email address, the IP address of the commenter will also be displayed. If you click this link, it will reveal all the comments from specific address.

In the Comment column, every comment involves certain information on the *Submitted on* followed by the time and date the comment was posted on the website. Clicking the time and date link will redirect you to that comment on your web site. When you over on a comment, you will see what you want to do with that comment: approve, reply, edit, trash, or mark as spam.

There are three elements in the column: In Response To . The text refers to the name of the post, which the comment is linked to as well as the links to the post editor for that particular entry.

Bonus: Claim Your Free Bonus Books Now!

I’d like to thank you for taking time to read my book. As a token of my graditude I’d like to offer you some of my #1 Best Seller Books for FREE!

You will also receive lots of great & free content in the future as well!! Simply click the link below and enter your name and email address to get your FREE content today!

Download Your FREE Books