

Skills and Topics for KidCoder: Beginning Web Design

Our Self-Study Approach

Our courses are **self-study** and can be completed on the student's own computer, at their own pace. You can steer your student in the right direction with no prior programming knowledge. Students only need typical computer usage skills to start; we will teach them programming from the ground up!

Each course comes with student activity starters, supplementary instructional documents, a **Solution Guide**, fully coded solutions for all activities, tests and answer keys, and guidance on evaluating projects.

Most questions about how to code individual activities are easily answered by referring to the Solution Guide (with or without parental involvement). We also provide **free technical support** to assist with any aspect of the courses!

Teachers who wish to closely monitor and grade student progress for credit purposes can administer chapter tests which are provided (with answer keys). We also provide advice and guidelines for evaluating student activities.

What Skills do Students Need to Begin?

All of our courses assume the student is already familiar with using a keyboard and mouse to select and run software, navigate the menus in a typical software program, and generally interact with their computer.

Students should understand how to use the built-in operating system software (Windows Explorer or Mac Finder) to find, save and retrieve files on their computer. It may also be helpful to have some familiarity with text editors (like Notepad or TextEdit) and some experience using web browsers to find information on the Internet.

We teach students how to create web pages from the ground up, but they should already know the basics about using a computer!

This course requires a **Windows or Mac OS** computer with a CD-ROM or DVD-ROM drive.

Topics Covered In This Course

The following are some of the web design topics that are covered in this course. For a full list of topics and sections, please see the Table of Contents for this course.

- Mark-up concepts
- Website layouts and files
- Backing up projects
- Essential HTML symbols
- Styling of text
- Using symbols and lists
- Internal and external hyperlinks
- Navigation bars and footers
- Simple CSS effects
- Spacing and positioning
- Graphics and image editing
- Tables



KidCoder™ Series

Beginning Web Design

A hands-on introduction to web design with HTML for elementary through high school students.



Student Textbook

First Edition

Copyright 2013 Homeschool Programming, Inc.

KidCoder™ Series



KidCoder™: Beginning Web Design

Student Textbook

First Edition

Copyright 2013

Homeschool Programming, Inc.

Thanks to Kathy Steffan for
helping us put this course together.

KidCoder™: Beginning Web Design

Copyright © 2013 by Homeschool Programming, Inc.

980 Birmingham Rd, Suite 501-128, Alpharetta, GA 30004

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means without written permission of the author.

ISBN: 978-0-9830749-6-0

Terms of Use

This course is copyright protected. Copyright 2013 © Homeschool Programming, Inc. Purchase of this course constitutes your agreement to the Terms of Use. You are not allowed to distribute any part of the course materials by any means to anyone else. You are not allowed to make it available for free (or fee) on any other source of distribution media, including the Internet, by means of posting the file, or a link to the file on newsgroups, forums, blogs or any other location. You may reproduce (print or copy) course materials as needed for your personal use only.

Disclaimer

Homeschool Programming, Inc., and their officers and shareholders, assume no liability for damage to personal computers or loss of data residing on personal computers arising due to the use or misuse of this course material. Always follow instructions provided by the manufacturer of 3rd party programs that may be included or referenced by this course.

Contact Us

You may contact Homeschool Programming, Inc. through the information and links provided on our website: <http://www.HomeschoolProgramming.com>. We welcome your comments and questions regarding this course or other related programming courses you would like to study!

Other Courses

Homeschool Programming, Inc. currently has two product lines for students: the *KidCoder™* series and the *TeenCoder™* series. Our *KidCoder™* series provides easy, step-by-step programming curriculum for 4th through 12th graders. These courses use readily available software products that come shipped with the operating system or are free to install in order to teach introductory programming concepts in a fun, graphical manner. Our *TeenCoder™* series provides introductory programming curriculum for high-school students. These courses are college-preparatory material designed for the student who may wish to pursue a career in Computer Science or enhance their transcript with a technical elective.

3rd Party Copyrights

This course runs on Microsoft Windows or the Apple Mac operating system. We demonstrate web pages using a variety of web browsers such as Internet Explorer, Mozilla Firefox, Google Chrome, and Apple Safari. All trademarks and copyrights relating to those operating systems and web browsers belong to their respective companies.

Instructional Videos

This course may be accompanied by optional Instructional Videos. These Flash-based videos will play directly from a DVD drive on the student's computer. Instructional Videos are supplements to the Student Textbook, covering every chapter and lesson with fun, animated re-enforcement of the main topics.

Instructional Videos are intended for students who enjoy a more audio-visual style of learning. They are not replacements for the Student Textbook which is still required to complete this course. However by watching the Instructional Videos first, students may begin each textbook chapter and lesson already having some grasp of the material to be read. Where applicable, the videos will also show "screencasts" of a real programmer demonstrating some concept or activity within the software development environment.

This Student Textbook and accompanying material are entirely sufficient to complete the course successfully. Instructional Videos are optional for students who would benefit from the alternate presentation of the material. For more information or to purchase the videos separately, please refer to the product descriptions on our website: <http://www.HomeschoolProgramming.com>.

Table of Contents

Terms of Use	3
Disclaimer	3
Contact Us	3
Other Courses	3
3 rd Party Copyrights	3
Instructional Videos	4
Table of Contents	5
Before You Begin	11
Minimum Hardware and Software Requirements	11
Conventions Used in This Text.....	12
What You Will Learn and Do In This Course.....	13
What You Need to Know Before Starting	13
Software Versions	13
Course Errata.....	13
Getting Help.....	14
Activity Starters	14
Chapter One: Introduction to Web Design.....	15
Lesson One: How the Internet Works.....	17
Lesson Two: Web Browsers.....	21
Lesson Three: The Language of “Mark-up”	27
Lesson Four: Ways to Create HTML	30
Chapter Review	33
Your Turn Activity: Secret Message Hunt.....	34
Chapter Two: Web Site Files and Directories	35
Lesson One: File Naming and Extensions	35
Work with Me: Make File Extensions Visible.....	38

KidCoder™: Beginning Web Design

Lesson Two: Web Pages	40
Lesson Three: Starting a New Web Site.....	43
Work with Me: Create the “Raptors” Folder	45
Work with Me: Back It Up!.....	47
Chapter Review	48
Your Turn Activity: Raptors Site Map	48
Chapter Three: Your First Web Page	49
Lesson One: Essential HTML Symbols.....	49
Lesson Two: HTML File Layout.....	52
Work with Me: Create Your Raptors Home Page.....	53
Work with Me: Add the <head> element	54
Work with Me: Add the <body> Element.....	54
Work with Me: Saving and Viewing a Web Page.....	55
Problem Solving Hints.....	58
Lesson Three: Header Elements	59
Work with Me: Adding <meta> Tags.....	60
Work with Me: Adding <!DOCTYPE> to Your Home Page.....	62
Work with Me: Add a <title> Element to the Home Page	64
Problem Solving.....	65
Lesson Four: Body Elements.....	66
Work with Me: Add a <h1> Heading to the Raptors Home Page	67
Work with Me: Adding Paragraphs.....	70
Problem Solving.....	71
Chapter Review	72
Your Turn Activity: Word Games	73
Chapter Four: Web Content	75
Lesson One: Highlighting Content.....	75
Work with Me: Bold Words on the Raptors Home Page	76

Table of Contents

Lesson Two: Dividing Up Your Page	80
Work with Me: Adding <div> Tags to Your Home Page.....	82
Lesson Three: Keeping It Neat	83
Work with Me: Cleaning Up Your Page.....	87
Work with Me: Adding Footer Content.....	88
Lesson Four: HTML Lists.....	90
Work with Me: Creating an Ordered List	91
Chapter Review	93
Your Turn Activity: Make an Unordered List.....	94
Chapter Five: Connecting Your Site	95
Lesson One: Hyperlinks	95
Work with Me: Add an Anchor Bookmark to the Home Page	97
Work with Me: Add a Link to Your Bookmark Anchor.....	98
Lesson Two: The Internal (Relative) Link	99
Your Turn Activity One: Practicing File Paths.....	102
Lesson Three: The Navigation Bar	102
Work with Me: Build Your Navigation Bar.....	104
Lesson Four: The External (Absolute) Link.....	105
Work With Me: Add a Safer Email Link and External Links.....	110
Chapter Review	111
Your Turn Activity Two: Finish Linking Your Home Page.....	112
Chapter Six: Cascading Style Sheets.....	113
Lesson One: Inline Styles	113
Work with Me: Create an Inline Style.....	117
Lesson Two: Choosing Colors	118
Lesson Three: Embedded CSS	121
Chapter Review	124
Your Turn Activity: Embedded Raptors Styles	125

Chapter Seven: Practical CSS.....	127
Lesson One: External Cascading Style Sheets.....	127
Work with Me: Create and Link an External Style Sheet.....	129
Problem Solving Techniques	130
Lesson Two: Font Styles.....	131
Work with Me: Add To Your Style Sheet.....	136
Lesson Three: Comments and Spacing.....	137
Work with Me: Comment Your CSS.....	137
Work with Me: Setting Margins and Padding.....	140
Lesson Four: Styling by ID, Class, or Group.....	141
Work with Me: Grouping Headlines	143
Chapter Review	145
Your Turn Activity: Raptors Margins and Padding.....	146
Chapter Eight: Understanding Cascades.....	147
Lesson One: Cascading Explained.....	147
Lesson Two: More Cascading Patterns	150
Lesson Three: Styling Links	152
Work With Me: Changing Link Color.....	152
Work With Me: Adding Link States.....	154
Lesson Four: Borders.....	155
Work With Me: Adding Borders	156
Work With Me: Adding a Headline Border.....	158
Chapter Review	159
Your Turn Activity: Creating Borders for the Raptors Sidebar	160
Chapter Nine: Positioning.....	161
Lesson One: Arranging Containers.....	161
Work With Me: Understanding Absolute Positioning.....	165
Lesson Two: Positioning Nested Elements.....	167

Work With Me: Absolute Positioning with Nested Elements..... 167

Lesson Three: Playing With Layers..... 170

Work With Me: Understanding Layers..... 170

Lesson Four: Relative Positioning..... 174

Work With Me: Relative Positioning Practice..... 175

Work With Me: Overflow Practice 178

Chapter Review 180

Your Turn Activity: Positioning Raptors Containers 181

Chapter Ten: Branding Your Site..... 183

Lesson One: Creating the Brand 183

Work With Me: Styling id="banner" 184

Lesson Two: Background Images 186

Work With Me: Understanding Background Images..... 189

Lesson Three: Create Custom Bullets 192

Work With Me: Styling Raptors Lists..... 192

Work With Me: Adding a Custom Bullet to Raptors..... 193

Lesson Four: Styling Navigation Bars 194

Work With Me: Styling the Raptors Navigation Bar..... 195

Chapter Review 198

Your Turn Activity: Branding Raptors..... 198

Chapter Eleven: Working with Graphics 199

Lesson One: Finding Images 199

Lesson Two: Basic Image Editing..... 201

Your Turn, Activity One: Cropping and Resizing the Great Grey Owl..... 202

Lesson Three: Adding Sub-Pages and Photos 203

Work With Me: Making the Great Gray Owl Page..... 203

Work With Me: Adding the Great Grey Owl Picture..... 207

Lesson Four: Positioning and Styling Photos 208

KidCoder™: Beginning Web Design

Work With Me: Styling Raptors Photos.....	208
Chapter Review	211
Your Turn, Activity Two: Create Other Raptor Sub-Pages.....	212
Chapter Twelve: Tables	213
Lesson One: Adding a Table	213
Problem Solving for Tables.....	215
Work With Me: Adding the Raptors Comparison Table	215
Lesson Two: Table Formatting	218
Work With Me: Styling Your Raptors Comparison Table	219
Lesson Three: Finishing Touches for Tables	221
Work With Me: Captioning the Raptors Comparison Table.....	222
Chapter Review	225
Your Turn Activity: Comparing Raptor Sizes.....	226
Chapter Thirteen: Final Project	227
Your Turn, Activity One: Content Brainstorming.....	227
Your Turn, Activity Two: Design Your Page	228
Your Turn, Activity Three: Laying the Foundation.....	229
Your Turn, Activity Four: Graphical Wizardry	229
Your Turn, Activity Five: Construction Time	230
Your Turn, Activity Six: High-Styling	230
What's Next?	231
Index.....	233

Before You Begin

Please read the following topics before you begin the course.

Minimum Hardware and Software Requirements

This is a hands-on web design course! You will be writing HTML, reviewing supplemental course material, and working with files on your computer. Your computer must meet the following minimum requirements in order to successfully complete the assignments.

Computer Hardware

Your computer must meet the following minimum specifications:

	Minimum
CPU	1.6GHz or faster processor
RAM	1024 MB
Display	1024 x 768 video card
Hard Disk Size	3GB available space
DVD Drive	DVD-ROM drive

Operating Systems

In order to install the course software, your computer operating system must match one of the following:

Windows XP (x86) with Service Pack 3 or above
Windows Vista (x86 and x64) with Service Pack 2 or above
Windows 7 (x86 and x64)
Windows 8 (all versions except RT)
Mac OS X 10.5.8 or greater

Supported Web Browsers

You can use nearly any web browser on any computer system to view HTML. However some browsers display HTML differently. We have tested course material on current versions of the following browsers:



Internet Explorer



Mozilla Firefox



Google Chrome



Apple Safari

Conventions Used in This Text

This course will use certain styles (fonts, borders, etc.) to highlight text of special interest.

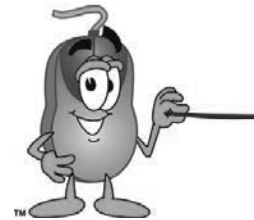
HTML source code will be in 11-point Consolas font, in a single box like this.

Property names will be in **12-point Consolas bold** text. For example: **#content{}**.

HTML elements and important terms will be in **bold face** type such as **<body>**.



This picture highlights important concepts within a lesson.



Sidebars may contain additional information, tips, or background material.



A chapter review section is included at the end of each chapter.



Every chapter includes a “Your Turn” activity that allows you to practice the ideas you have learned.



The “Work With Me” sections will give you step-by step instructions on how to apply the material to your project. Work alongside the instructions on your computer to achieve a goal.

Many “Work With Me” and “Your Turn” activities will ask you to add new code, edit existing code, or remove old code. We will use a light gray color to represent old code and black text for new code. Any existing code that needs to be removed will be crossed out.

```
This line shows work that is already in the file
This black text shows the work you need to add.
This line shows another row of text that was already in the file.
Crossed out text needs to be removed.
```

What You Will Learn and Do In This Course

KidCoder™: Beginning Web Design will teach you the basics of HTML, XHTML and CSS. It is written for students in 4th grade or higher who have an interest in building web sites. You will learn to create your own web pages and begin to understand the building blocks for developing other web sites!

Each lesson will include an explanation of concepts, examples of how concepts are used, and one or more activities that will help you understand the concept. Throughout the course, you will be challenged to apply what you have learned by building your own web site from scratch.

What You Need to Know Before Starting

You are expected to already know the basics of computer use before beginning this course. You need to know how to use the keyboard and mouse to select and run programs, use application menu systems, and work with either the Microsoft Windows or Apple Mac operating system. You should understand how to save and load files on your computer and how to use the Windows Explorer or Mac Finder to walk through your file system and directory structures. You should also have some experience with using text editors, like NotePad or TextEdit and web browsers, like Safari or FireFox.

Software Versions

You will be using either *Notepad* (PC) or *TextEdit* (Mac) software to complete this course. These programs come shipped with your operating systems. *Microsoft Paint* or *Mac Preview/iPhoto* are used in the graphics chapter, both of which also come shipped with your operating systems. All supplemental documents installed with the course material are in Adobe Acrobat (PDF) format. You must have the Adobe Acrobat Reader installed to view these documents.

Course Errata

We welcome your feedback regarding any course details that are unclear or that may need correction. You can find a list of course errata for this edition on our website.

Getting Help

Throughout the course you will be given some problem solving tips to help you find and fix problems. The earlier tips can be used to troubleshoot later exercises as well.

All courses come with a Solution Guide and fully coded solutions for all activities. Simply install the “Solution Files” from your course setup program and you will be able to refer to the solutions as needed from the “Solution Menu”. If you are confused about any activity, this will allow you to see how we solved the problem!

We also offer free technical support for students and teachers. Simply fill out the help request form in the “Support” area of our website with a detailed question and we will assist you.

Activity Starters

Some exercises and assignments require graphical images. We have provided all of those images for you in the “KidCoder/BeginningWebDesign/Activity Starters” directory. In addition, a few activities may require a lot of typing to enter text content. To make your job easier, we have also provided text files in the “Activity Starters” directory containing this text content. You can cut and paste from the starter text files into your own code to save some time. Please look at the “Activity Starters” tab in your Student Menu for details on the starter material. When starter material is available, it will be noted in the activity description.

Support for Multiple Operating Systems

This course was developed for use both on Microsoft Windows and Apple Mac OS X operating systems. While HTML can be used on nearly any computer platform, our course setup program will only run on these systems, and we give guidance using the tools and terms specific Windows and Mac OS. We will point out in text or by screen shots any differences between the operating systems. Where necessary, we will provide dedicated sets of instructions for handling each operating system. Be sure to follow the instructions that match the operating system you are using!

Directory Naming Conventions

On Windows operating systems, directory paths are traditionally represented with backslashes (“\”) between folder names like this: “**KidCoder\BeginningWebDesign**”. However, forward slashes (“/”) also work. On Mac OS, directories use forward slashes as in “**KidCoder/BeginningWebDesign**”. In order to avoid cluttering the textbook with both representations, each time we specify a path, we will simply use the forward slash (“/”) style which works on both operating systems.

SAMPLE STUDENT LESSON

**The following pages contain a sample student lesson from
the KidCoder: Beginning Web Design textbook.**



Chapter Three: Your First Web Page

At this point, you've prepared your computer to make your website development easier and more organized. Now it's time to start learning some HTML! In this chapter we will discuss the most important HTML symbols and describe the parts of a standard web page.

Lesson One: Essential HTML Symbols

A web page contains a mixture of both data and HTML mark-up. The mark-up is identified by some special symbols and tags that you will learn to use.

Angle Brackets

The most important symbols in HTML coding are the less-than (<) and greater-than (>) signs. These symbols, when combined with words in between, tell the web browser how to display the data on the web page. The (<) and (>) signs are called **angle brackets**, or just "**brackets**" for short in HTML.

Tags

When angle brackets are used around a specific word, it is called a **tag**. Tags are used to mark the start and end of sections of a web page and will tell the browsers how to display those sections.

Imagine each bracket is the mouth of a crocodile. Crocodiles open their mouths to capture their prey. They are greedy creatures, so there are always two crocodiles for every prey. One grabs from the front, and the other from the back, like this: <PREY>. When the brackets are positioned this way around a word, it marks the start of the tag. In HTML, this is called the **opening tag**.



At some point, the tag needs to end, or stop. This is done by adding a **forward slash (/)** before the name of the tag, like this: </PREY>. The HTML term for this type of tag is the **closing tag**.

Let's go back to our crocodile example. Our crocodiles will hold on to their prey until a zookeeper comes with a big stick and pokes the first one in the mouth. The zookeeper doesn't want to stand on the

KidCoder™: Beginning Web Design

crocodile's sharp teeth or in the water to stop the crocodiles, so he stands on top of the prey to get the right angle as he pokes at the first crocodile. If you can remember this scene, you will always know to use the forward slash (/) and not the backward slash (\) to begin a closing tag. Only the first crocodile needs to be poked and they both let go of their prey.

In an HTML web page your data is surrounded by an opening tag and then the closing tag:

```
<PREY>your data</PREY>
```

The tag word will tell the web browser exactly how to display your data. Now, <PREY> is not actually an HTML tag, so don't try to use it on your web site! You will learn some real HTML tags shortly.

HTML Elements

When tags are put together with a start, middle and end they are called an **element**. An HTML element is everything from the start tag to the end tag, including the tags themselves and the data within.

Each element is like a container or box. Some are really large and some are very small. Have you ever played with those little wooden dolls that nest inside each other? To organize them correctly, you start with the largest doll and then place each smaller doll inside, until they are all stacked neatly inside the first doll. Web elements work in a similar way. The largest container (or element) is put down first and then all the others sit inside this container. You will be learning how many of these elements sit within and beside each other to make your web site work.

Attributes and Values

Many tags can have extra pieces of information attached to them, called **attributes** and **values**. These extra pieces of information are located inside an opening HTML tag (before the end bracket >).

Attribute and value pairs are often used to define how the tag looks or behaves on your web page. These pairs can contain numbers, words, link locations, or other types of data. In your HTML code, an attribute name is followed by the equal sign and then a value surrounded by quotation marks.

```
<tag attribute="value">your data</tag>
```

You will learn more about specific attributes and values and how to use them later in this course.

In XHTML, the value must always be surrounded by double quotes. Make sure you use the straight double quotes (" "), and not two single quotes beside each other (' ') or those fancy curly quotes (“ ”). The value will only work correctly if you use straight double quotes.

Older HTML code did not care what type of quotation marks you used or even if you used any at all! It wasn't until XHTML came along that the rules became more rigid. However, these stricter rules actually allow different web browsers to better understand your web page.



The new HTML5 standard allows some flexibility in the use of quotes and values. However, it is best to stick with the XHTML style until HTML5 is universally approved and accepted, which will take a few years!

In the original version of HTML, it didn't matter if you used capital or lower case letters for your tag names. So, `<PREY>` and `<prey>` would do the same thing. In XHTML, however, only lower case letters are used for tags. For this course, make sure to use all lower case tag names to match the XHTML standard.

Empty Elements

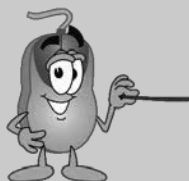
Most elements have a starting tag, content in the middle, and a closing tag. However, some tags will never have any data inside. One example is the line break element “`br`”. This tag is simply used to force the text in a web page to break and move to the next line. Instead of writing both closing and ending tags like “`
</br>`”, you can use an “empty” or “self-closing” format.

With empty elements, the tag starts like usual, but instead of a separate closing tag, there is a space and a forward slash just before the end bracket. So our empty line break element “`br`” would be written like this:

```
<br />
```

The original rules for HTML were not very well defined. This allowed for lazy and sloppy web page code. Using the original HTML, if a web designer started a tag, and then forgot to close it, the web browser would try to figure out what was meant and display the web page as best it could. This made life easier for the web designers, but made a web browser's job much more difficult.

The XHTML standard changed things dramatically. XHTML requires all HTML tags to be closed either by using a closing tag or, for empty elements, adding a forward slash inside the start tag. Be sure to always close your starting tags with ending tags or use empty tags that are self-closed in your own HTML code!



**There is a debate about putting a space before the slash in an empty tag. When XHTML first came out, browsers didn't recognize `
` as a shortcut for `
</br>` and would ignore it. But if you added a space before the end bracket as in “`
`” the web browsers would process the element correctly. Since this is a common practice, our examples will follow that format.**

Lesson Two: HTML File Layout

In this lesson, you will build your first web page! Every web page has one `<html>` root element that contains all other tags and data. Inside the `<html>` root are two main sections: the `<head>` and the `<body>`.

```
<html>
  <head>
  </head>
  <body>
  </body>
</html>
```

Every web page should have these three basic elements. Notice how the `<html>` element contains the other two, with the ending `</html>` tag at the very end. Let's take a closer look at these three elements.

The `<html>` Element

The opening `<html>` tag is always the first tag in your web page and the closing `</html>` tag is always the last tag in the page. These HTML tags will tell the browser that everything between these two tags is the actual content of our web page.

```
<html>
</html>
```

How does the web browser know if you are going to be using HTML, XHTML or some other standard? We'll add an attribute and value to the `<html>` tag so the browser will understand the type of markup on the page. Most of our pages in this course will use the XHTML standard because it's good practice to follow the strict XHTML coding rules.

To mark the web page as XHTML, add an attribute named `"xmlns"` with a value of `"http://www.w3.org/1999/xhtml"` to the `<html>` opening tag.

```
<html xmlns="http://www.w3.org/1999/xhtml">
</html>
```

Follow the instructions in the next section in order to create your first web page!



Work with Me: Create Your Raptors Home Page

1. Open your Notepad (Windows) or TextEdit (Mac) text editor
2. On the blank page type the opening HTML element and attribute, using lower case letters

```
<html xmlns="http://www.w3.org/1999/xhtml">
```

Use the ENTER key ↵ to move your cursor down the page several lines

```
↵  
↵  
↵  
↵  
↵
```

3. At the end of your blank lines, close the HTML element with the closing tag:

```
</html>
```

4. Keep this file open, because you'll add some more HTML code to it in a minute!

The <head> Element

The first element inside the root **<html>** is the **<head>** element. This is the “brain” of the page. The **<head>** area contains information about the page and instructions for the browser, but is not visible to people reading the page in the browser. You will learn some of the specific tags that go in this area later. For now you can set up an empty **<head>** section in your web page. Follow the instructions in the next section to add your **<head>** element now.



Work with Me: Add the <head> element

1. In the same file you have open from the last section, move your cursor to the line below the starting <html> tag and type in your starting <head> tag.

```
<html xmlns="http://www.w3.org/1999/xhtml">  
<head>
```

2. Use the ENTER key ↵ to move your cursor down the page 2 lines.

```
↵  
↵
```

3. Now type the closing </head> tag before the </html> tag.

```
</head>  
</html>
```

4. Keep this file open, because you'll add the <body> element next!

The <body> Element

The <body> element comes right after the closing </head> tag, and contains everything you see on the screen. All text, images, links to other pages, etc. is stored in the body section. Let's add the <body> element to your new web page now.



Work with Me: Add the <body> Element

1. In your open file, move your cursor to the line below the closing </head> tag and type in the opening <body> tag.

```
</head>  
<body>
```

2. Use the ENTER key ↵ to move your cursor down the page 2 lines

```
↵  
↵
```

3. Now type the tag needed to close the body.

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

4. Keep this page open. We're done adding HTML tags for now, and we'll save it next.

You now have a basic web page with the three main elements. We started with the opening `<html>` tag, which is the biggest container for the page. Then we placed the `<head>` section at the top of the `<html>` section. The `<body>` container also fits inside the `<html>` container and is placed below the `<head>`. These three tags are all you need to call your file a web page! Now it's time to save the open file to disk and take a look at it in a web browser.

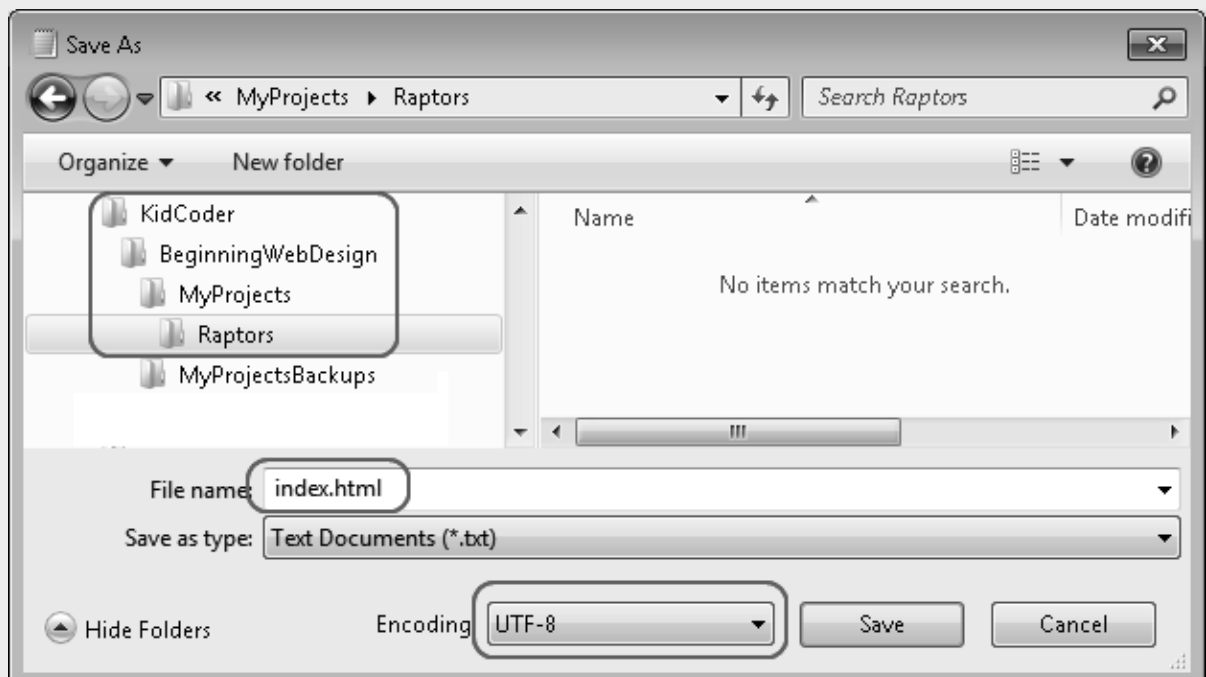


Work with Me: Saving and Viewing a Web Page

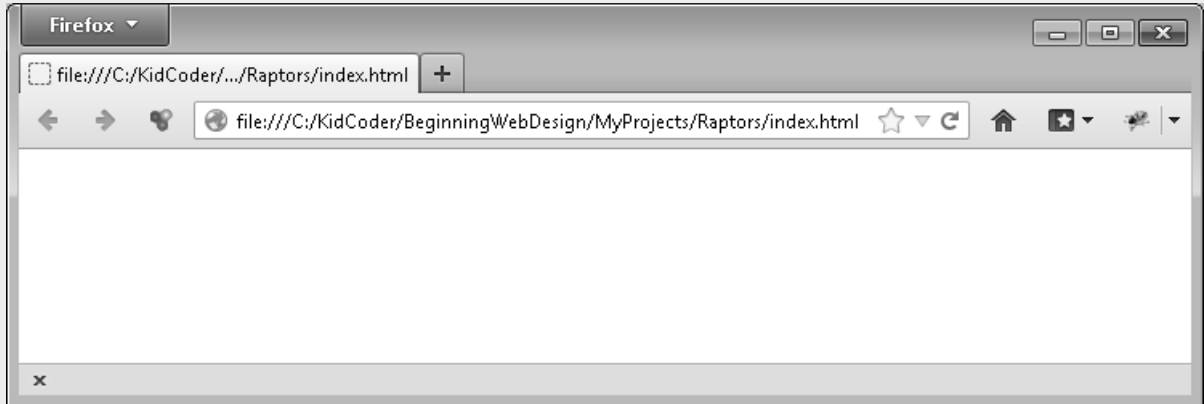
Let's save your new web page as an ".html" file. If a file has an ".html" extension on it you can double-click that file in your Windows Explorer or Mac Finder and it will be automatically opened by your default web browser and shown as a web page.

Windows Instructions

1. Select "File → Save As" from the Notepad menu.
2. In the "Save As" dialog, move to your "KidCoder/BeginningWebDesign/MyProjects" folder and select the "Raptors" subdirectory, which is currently empty.
3. Change the "File name" field to contain "**index.html**" and select the "**UTF-8**" option from the "Encoding" combo box. Make sure you manually add the **.html** extension when you type the file name or your text editor may default it to a ".txt" file. Click the "Save" button when done.

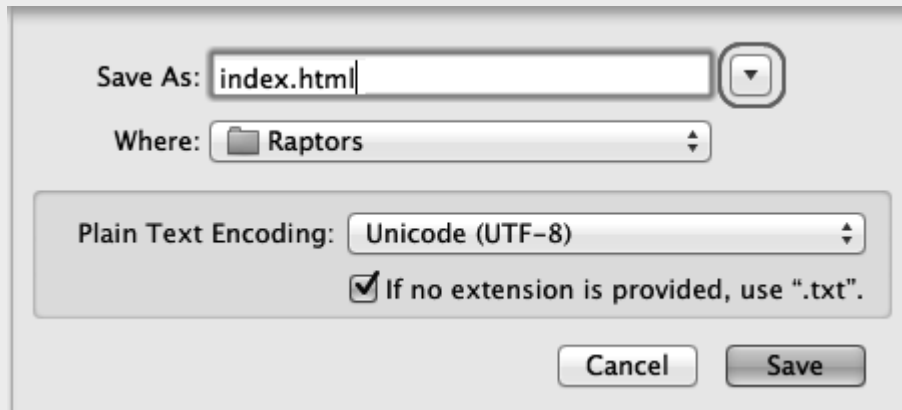


4. Now that your file is saved, run Windows Explorer and navigate again to your “Raptors” folder. You should see your new “index.html” file within this folder.
5. Double-click the file name. Your default web browser should open in your default web browser and show a blank white page.

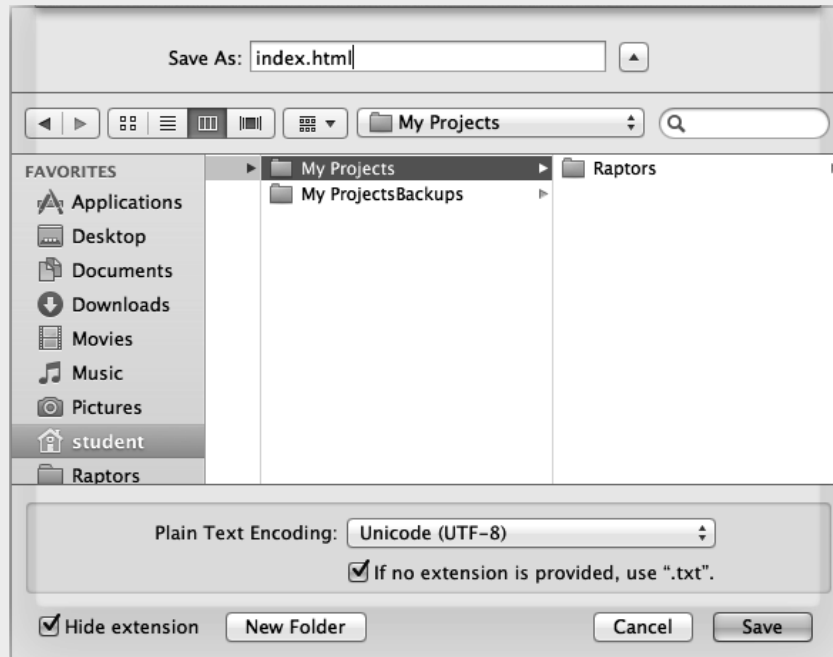


Mac Instructions

1. Use the Finder “Menu” bar to select “Format”
2. From the drop-down list, select “Make Plain Text”
3. Find the menu at the top of the window and click “File”
4. From the menu that shows up, click “Save”
5. Beside “Save As”, change the word “Untitled” to “index.html”. Make sure you add the .html extension when you type the file name or TextEdit may give it a .txt extension.

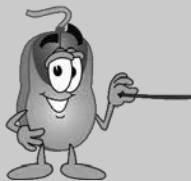


6. If you have moved the “Raptors” folder to your Finder sidebar, you can use the “Where” drop-down list to select your “Raptors” web site folder.
7. If you did not move your “Raptors” folder to the Finder sidebar, you will need to click the down arrow next to the “Save As” box. This will allow you to navigate to the correct folder on your computer.



8. Make sure “Plain Text Encoding” is set to “UTF-8”
9. Click “Save” to save your file.
10. If a box appears that warns you about the use of the extension “.html”, just click on the button labeled “Use .html”
11. Use Finder to open the “Raptors” folder. Inside it, you should find your file “index.html”.
12. Double-click the file name. Your default web browser should open in your default web browser and show a blank white page.

When you finish working on your “index.html” file for the day, you can close your Notepad or TextEdit program. Later you can re-open that same file and continue working on it. To re-open a file, first run Notepad or TextEdit as usual, and then select “File → Open” from the top menu. Browse to your “My Projects/Raptors” directory and select the “index.html” file to open it again.



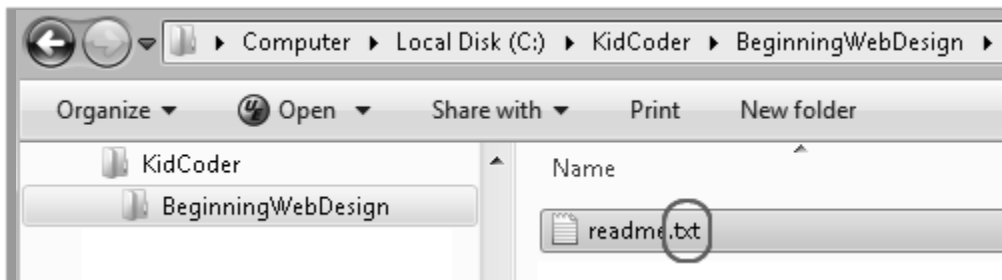
If you are using a Mac computer, you may need to make a simple change to the configuration of the TextEdit program in order to view the HTML code instead of the formatted web page. Please click on the “Installation and Extras” tab from your Student Menu and follow the link to our online documents page. There you will find detailed instructions for this small change.



Problem Solving Hints

If your web page file did not open properly in your web browser, check these things first.

1. Do you find the right file? You should be looking for a file named: “/KidCoder/BeginningWebDesign/MyProjects/Raptors/index.html”
2. Did you remember to show the file extensions on your computer? If you are not sure, use Windows Explorer or Mac Finder to look in your “/KidCoder/BeginningWebDesign” folder and see if you can view the file extensions in that folder. You should see “readme.txt” or “readme.rtf”, for example, instead of just “readme”.



If you can't see any extensions, go back to the instructions on showing file extensions in Chapter Two and try again.

3. Was your file named correctly? Your text editor may have named the file “**index.html.html**” or “**index.html.txt**” accidentally. If so just rename your file to remove the incorrect extension.
4. Did you misspell the “.html” extension? If the extension is “.html” or “.hmlt” or something other than “.html”, you will need to rename the file to correct the extension.
5. Did you make any mistakes in your mark-up code? Is anything spelled wrong? Did you miss an angle bracket or opening tag or closing tag? Did a comma (,) or period (.) show up where you thought you put a bracket? Did you forget the forward slash to turn off a tag? Did you miss a tag entirely?

The Solution Files installed on your teacher's computer contain fully coded solutions for all activities. If you can't spot the error, you can carefully compare your file to the answers within the Solution Files.

SAMPLE SOLUTION GUIDE

The following pages contain sample solution material for an activity in the KidCoder: Beginning Web Design textbook.

Chapter Seven Activity (Raptors Margins and Padding)

In this exercise your student is going to set your Raptors margins and add some padding around the edges. They will do this by adding a new `<div>` element to the “index.html” home page and adding some new CSS rules to the “global.css” style sheet.

Here is the code they need to add to the top of the “index.html” file (in bold text):

```
<body>  
<div id="edges">
```

And then the code needed at the bottom of the same file:

```
</div><!-- end of footer -->  
</div><!-- end of edges -->  
</body>
```

Next, your student will need to move to their “global.css” file and add the following style rules underneath the body section:

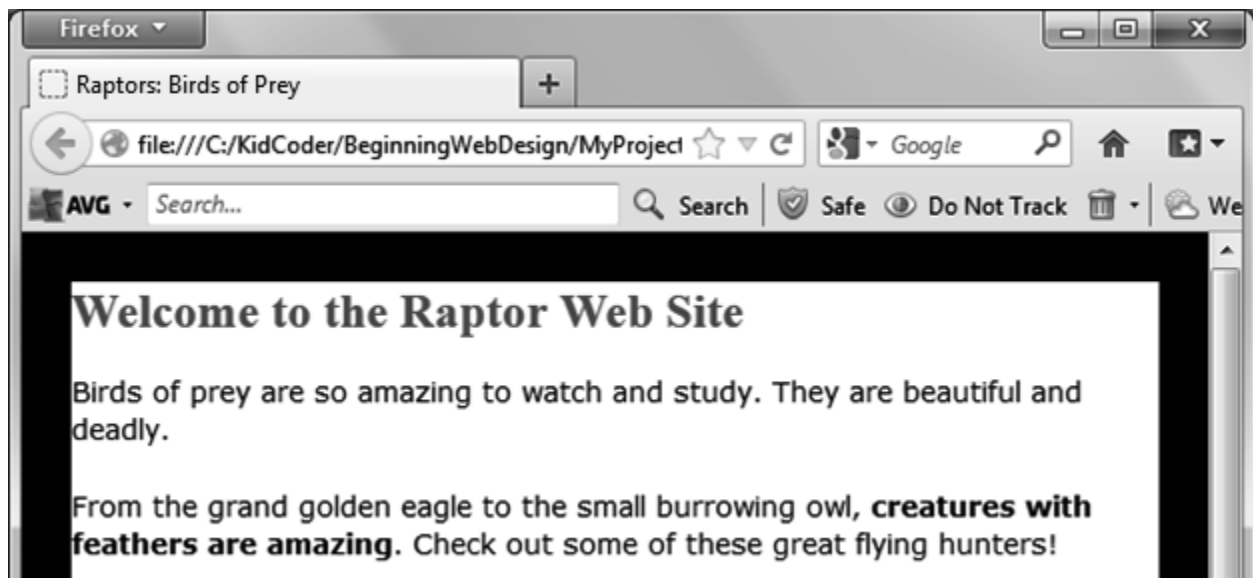
```
#edges{  
    background-color: #FFFFFF;  
    color: black;  
    margin: 25px;  
}  
  
p {  
    padding-bottom: 5px;  
}
```

Chapter Seven Activity (Raptors Margins and Padding)

Finally, the student should change the existing body `<body>` rule to switch the **background-color** and **color** properties.

```
body {  
    background-color: black;  
    color: white;  
    font-family: Verdana, Helvetica, Arial, sans-serif;  
    font-size: 0.9em;  
    padding: 0;  
    margin: 0;  
}
```

When they are done, they should load their “index.html” file into a web browser:



You should see a black margin around the edge that is 25 pixels wide. The interior of the “edges” container should be white with black text and you should also see that the paragraphs have a bit of extra space between them.

Make sure your student makes a backup of their Raptors folder to their chosen backup location. Review “Making Backups” at the end of Chapter Two for a reminder of how to do this, if needed.

The completed HTML file for this activity is “Activity Solutions/Chapter07/index-assignment.html” located in the directory where you installed the “Solution Files”. Similarly, the completed CSS file is “Activity Solutions/Chapter07/SiteStyle/global-assignment.css”.