

Skills and Topics for KidCoder: Advanced 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.

KidCoder: Advanced Web Design is a second-semester course. **Students must have completed the KidCoder: Beginning Web Design first-semester course to learn the fundamental HTML skills necessary to start this course!**

Topics Covered In This Course

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

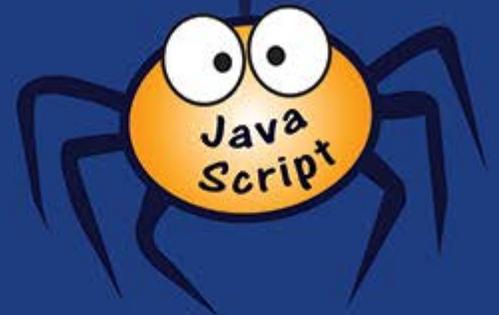
- Using Komodo Edit
- New HTML5 Elements and Syntax
- Headers, Footers, Articles, and Asides
- Dynamic Navigation Menus
- Figures and Dynamic Annotations
- Embedded Audio and Video
- New CSS3 Styles
- Advanced Borders and Shadows
- Using Custom Fonts
- Color Gradients and Image Transformations
- Advanced Relationship and Pseudo-Selectors
- Introduction to JavaScript
- Animation with jQuery



KidCoder™ Series

Advanced Web Design

A hands-on study of web design with HTML5, CSS, and JavaScript for elementary through high school students.



Student Textbook

First Edition

Copyright 2013 Homeschool Programming, Inc.

KidCoder™ Series



KidCoder™: Advanced Web Design

Student Textbook

First Edition

Copyright 2013

Homeschool Programming, Inc.

Thanks to Kathy Steffan for
helping put this course together.

KidCoder™: Advanced 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: **978-0-9887033-3-9**

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. Komodo Edit is copyright by ActiveState Software, Inc.

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 all you need 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.....	14
Getting Help.....	14
Activity Starters	14
Chapter One: Getting Started.....	15
Lesson One: What You Already Know	17
Lesson Two: Web Editing Tools.....	23
Your Turn Activity: Installing Komodo Edit	24
Lesson Three: Komodo Edit Projects.....	25
Work with Me: Turning off the Komodo Start Page.....	27
Work with Me: Creating the Aquamaniacs Project	30
Work with Me: Creating Aquamaniacs Home Page.....	33
Work with Me: Komodo Edit Browser Preview	36
Chapter Review	38

Your Turn Activity: Laying Out the Aquamaniacs Home page.....38

Chapter Two: Introducing HTML5.....39

Lesson One: HTML5 and Web Browsers.....39

Work with Me: Verify Your Web Browser.....42

Lesson Two: Identifying HTML5 Pages.....43

Work with Me: Adding HTML5 Meta Tags to the Home Page.....45

Lesson Three: HTML5 Element Updates.....46

Chapter Review.....50

Your Turn Activity: Using Updated HTML5 Elements.....50

Chapter Three: Bringing Meaning to Markup.....51

Lesson One: Defining Areas with Sections.....51

Lesson Two: Headers and Footers.....54

Work with Me: Using Header and Footer on the Home Page.....55

Lesson Three: Articles, Asides, and Navs.....57

Work with Me: The Aquamaniacs Navigation Bar.....59

Work with Me: Adding the Dolphins Page.....63

Chapter Review.....65

Your Turn Activity: Styling Articles and Asides.....65

Chapter Four: Figures and Annotations.....67

Lesson One: Figures and Captions.....67

Work with Me: Watery Pictures.....68

Lesson Two: Figure Annotation with Relative Positioning.....71

Work with Me: Aquarium Annotations.....75

Lesson Three: Dynamic Annotations.....77

Work with Me: Dynamic Fish Annotations.....80

Chapter Review.....83

Your Turn Activity: Leaping Dolphins.....84

Chapter Five: Adding Audio and Video	85
Lesson One: Linked and Embedded Videos.....	85
Lesson Two: HTML5 Video.....	89
Work with Me: Octopus Videos.....	91
Lesson Three: HTML5 Audio	93
Chapter Review	95
Your Turn Activity: Sound Bites.....	96
Chapter Six: Introducing CSS3	97
Lesson One: Browser Compatibility.....	97
Work with Me: Check Your Browser CSS3 Compatibility	100
Lesson Two: Colors and Transparency.....	101
Work with Me: Transparent Whales	103
Lesson Three: Advanced Borders and Shadows.....	106
Work with Me: Styling the MainContent <div>	108
Lesson Four: Custom Fonts.....	110
Work with Me: Adding “Sigmar” Font Headlines	114
Chapter Review	117
Your Turn Activity: Branding Aquamaniacs	118
Chapter Seven: Cool CSS3 Features	119
Lesson One: Advanced Backgrounds.....	119
Work with Me: Adding the Octopus Background	121
Lesson Two: Gradients.....	124
Work with Me: Groovy Fish Gradients	128
Lesson Three: Image Transformations	131
Work with Me: Preparing to Transform Funny Fish 1.....	131
Work with Me: Funny Fish 1 Transforms	136
Chapter Review	137

Your Turn Activity: Grooving and Shaking.....	138
Chapter Eight: Relationship Selectors	139
Lesson One: Parent-Child Relationships	139
Work with Me: Drawing a Node Tree.....	143
Lesson Two: Parent and Child Selectors	144
Work with Me: The Emperor Penguin	147
Lesson Three: Sibling Selectors	151
Work with Me: Distant Siblings on the Penguins Page	153
Chapter Review	154
Your Turn Activity: Emperor Penguins.....	154
Chapter Nine: Pseudo-Selectors	155
Lesson One: Pseudo-Class and Pseudo-Element Selectors.....	155
Work with Me: Playing with “:first-letter” and “:first-line”	158
Lesson Two: First and Last Child Pseudo-Selectors.....	160
Work with Me: Playing with “:first-child” and “:last-child”	161
Lesson Three: Numbered and Combined Selectors	163
Work with Me: Playing with Numbered Selectors	165
Work with Me: Adding a Drop-Cap Effect.....	167
Chapter Review	169
Your Turn Activity: Lobsters.....	169
Chapter Ten: Dynamic Menus.....	171
Lesson One: Dynamic Menus and Nested Lists	171
Work with Me: The Aquamaniacs <nav> Node Tree	174
Work with Me: Preparing the Aquamaniacs Header.....	175
Lesson Two: Styling First-Level Dynamic Menus.....	178
Work with Me: Starting the Aquamaniacs Drop-Down Menus.....	182
Lesson Three: Styling Second-Level Dynamic Menus.....	185

Work with Me: Finishing the Aquamaniacs Drop-Down Menus.....	189
Chapter Review	193
Your Turn Activity: Lionfish	193
Chapter Eleven: Introducing JavaScript	195
Lesson One: JavaScript Concepts	195
Work with Me: Your First JavaScript Commands.....	199
Lesson Two: JavaScript Events and Functions.....	201
Work with Me: Waking the Seahorse	205
Lesson Three: Managing Elements with JavaScript	206
Work with Me: Good Morning Sunshine	210
Lesson Four: External JavaScript Files.....	212
Chapter Review	216
Your Turn Activity: Cuttlefish.....	217
Chapter Twelve: Dynamic Pages with jQuery	219
Lesson One: The jQuery Library	219
Work with Me: Getting Started with Funny Fish 2.....	222
Lesson Two: jQuery Syntax and Page Loading.....	223
Work with Me: Funny Fish 2 Page Loading.....	227
Lesson Three: jQuery Events	229
Work with Me: Funny Fish 2 Page Events	233
Lesson Four: Hiding and Showing Elements.....	235
Work with Me: Funny Fish 2 Hide, Show, and Toggle	238
Chapter Review	240
Your Turn Activity: Otters.....	240
Chapter Thirteen: jQuery Special Effects	241
Lesson One: Fading and Sliding.....	241
Work with Me: Fading Funny Fish 2.....	243

Work with Me: Fading and Sliding Otter.....	245
Lesson Two: Timing and Multiple Effects.....	248
Work with Me: Finishing the Otter	250
Lesson Three: jQuery Animation.....	252
Work with Me: Octopus Animation.....	256
Chapter Review	260
Your Turn Activity: Dancing Octopus.....	260
Chapter Fourteen: Final Project.....	261
What's Next?	263
Index.....	265

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.

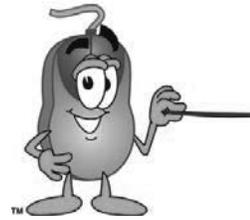
HTML source code will be in 11-point Consolas font, in a single box like this.

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™: Advanced Web Design starts where the *KidCoder™: Beginning Web Design* course left off. This course is written for 4th grade or higher school students who have an interest in building web sites. You will learn HTML5, CSS3, JavaScript, and other new features to make exciting and interactive web sites. You will also learn how to use a free program called Komodo Edit, which makes typing out code much easier.

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 have a good understanding of the material in *KidCoder™: Beginning Web Design*. If you have any questions about that course, please review the earlier topics before beginning this course.

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 the free, cross-platform *Komodo Edit* software to complete this course. You will be guided through the download and installation of this software during an early activity. You may also use *Microsoft Paint* or *Mac Preview/iPhoto* to create and edit graphics and images. These programs 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 can refer to the solutions as needed from the “Solution Menu”. If you are confused about any activity, you can 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/AdvancedWebDesign/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.

Directory Naming Conventions

On Windows systems, directory paths are traditionally represented with backslashes (“\”) between folder names like this: “**KidCoder\AdvancedWebDesign**”. However, forward slashes (“/”) also work. On Mac OS, directories use forward slashes as in “**KidCoder/AdvancedWebDesign**”. 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

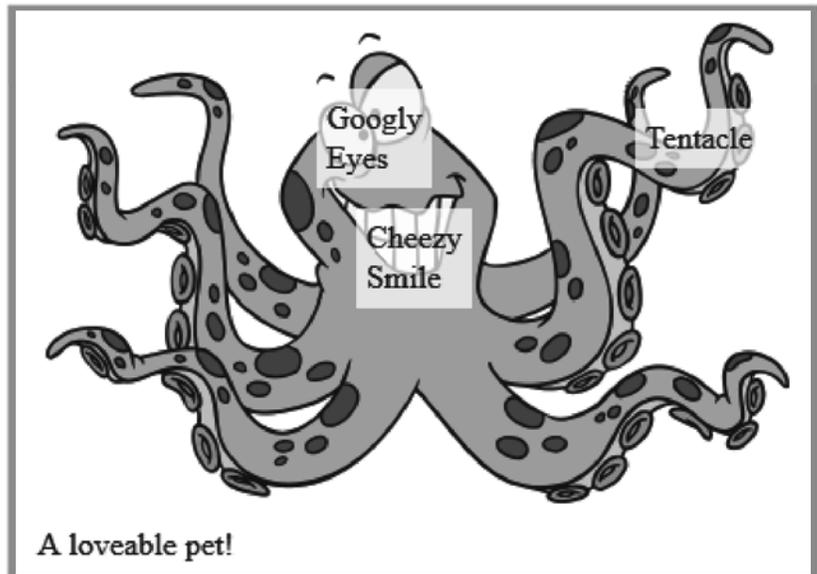
**The following pages contain a sample student lesson from
the KidCoder: Advanced Web Design textbook.**

Lesson Two: Figure Annotation with Relative Positioning

An **annotation** is a small note you can add to a picture. If you’ve ever “tagged” someone in Facebook, you have seen annotations in the form of boxes drawn around the faces of the people in your pictures. Annotations can be made up of simple text or fancier graphics.

In this example, we have added three annotations to our octopus: “Googly Eyes”, “Cheezy Smile”, and “Tentacle”.

These effects are pretty easy to create with HTML5 and CSS using these steps:



1. Use a special class of **<figure>** and an unordered list **** within the **<figcaption>** element. Each list item **** contains one annotation.
2. Apply inline styles to each **** with specific location information
3. Add style rules to your CSS to control the positioning and other special effects

Creating Annotations in a List

To create a nice annotation on an image, your first step is to ensure that you are using the HTML5 **<figure>** and **<figcaption>** elements. Add a **class** attribute to your **<figure>** element so we can easily create style rules applying to any annotated figure. We have chosen the class name “**annotated**”, but you can use any name you like. Then within your **<figcaption>**, add an unordered list **** with some list items **** containing each annotation or piece of text you want to show in the image.

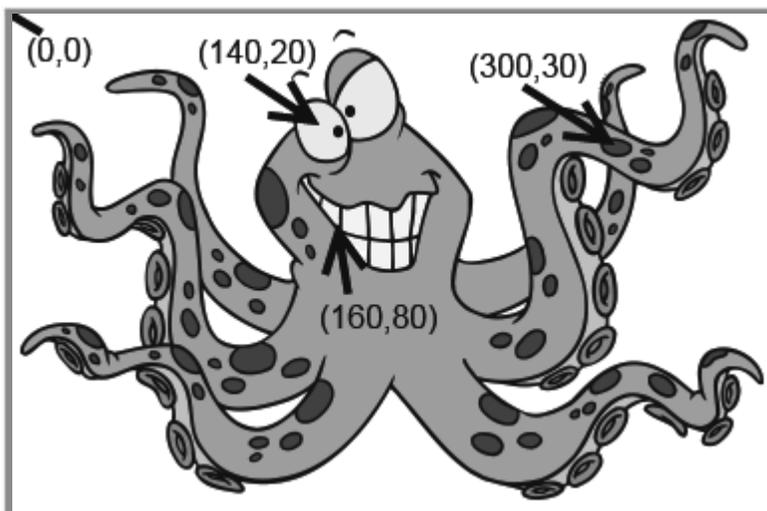
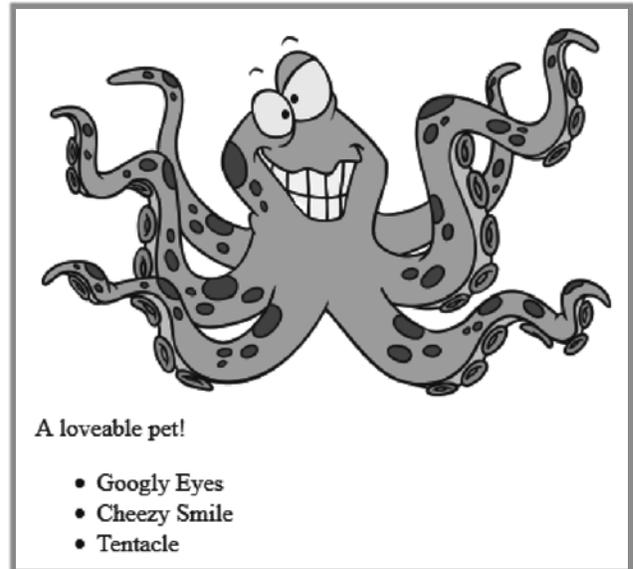
```
<figure class="annotated">
  
  <figcaption>A loveable pet!
    <ul>
      <li>Googly Eyes</li>
      <li>Cheezy Smile</li>
      <li>Tentacle</li>
    </ul>
  </figcaption>
</figure>
```

So far this extra list will just appear underneath the caption. But we really want to move each list item to a specific location on the image.

Inline Location Styles for List Items

In order to place each list item on the image in the right location, we need to know the coordinates for each spot. A **coordinate** is a pair of values that counts the number of pixels over and down from the top-left corner of the image.

The top-left corner of an image is written as coordinate (0,0). As you move pixels to the right, the first value increases, so (10,0) would be a pixel 10 places over from the left side. Similarly, as you move down the image, the second value increases. The coordinates (10,20) would point to a spot 10 pixels to the right and 20 pixels down from the top-left corner.



You can figure out coordinates for any spot on an image using an image editor. Usually just hovering your mouse over a spot will show the coordinates somewhere on the screen. Or you can just guess with some trial and error until you home in on the right spot. The image to the left shows the key locations for each annotation, plus the top-left (0,0) coordinate for reference.

Now that we know the locations for each piece of text, we want to set those locations using the CSS properties **left** and **top**. The **left** property will hold the number of pixels across from the left side, and the **top** property will hold the number of pixels down from the top edge. Because each list item will have different coordinates, we will apply these properties inline directly on each element instead of using a separate CSS file.

```
<ul>  
  <li style="left: 140px; top:20px;">Googly Eyes</li>  
  <li style="left: 160px; top:80px">Cheezy Smile</li>  
  <li style="left: 300px; top:30px">Tentacle</li>  
</ul>
```

Unfortunately just adding these top and left coordinates with styles is not enough; you won't see any visible change in our image! Location of elements on the screen can be done using *absolute* or *relative* positioning. We covered the **position** property in detail in our first-semester course, so you may want to look back on that material for a reminder.

In order to place the annotations in the right spots, we need to add three CSS rules. First, we want to add a rule for the “annotated” class that changes the **position** to *relative*. This style tells the browser that elements inside may be positioned somewhere other than their default positions.

```
.annotated
{
    position:relative;
}
```

Next we need to add a style rule for the unordered list `` inside the annotated figure. These properties should all be familiar to you from earlier work. We are removing the bullet icons from the list and setting the font size. The absolute position is very important, because it tells the browser to place the `` element exactly at the top and left positions shown (0,0) relative to the parent `<figure>`.

```
.annotated ul{
    list-style:none;
    position:absolute;
    top:0;
    left:0;
    font-size:16px;
}
```

Now we can start seeing some results in our browser! The list items appear near the top-left corner of the figure. We have successfully positioned the entire list element ``, but the individual locations for each `` are still being ignored. We will need to add a third style rule to make those take effect.



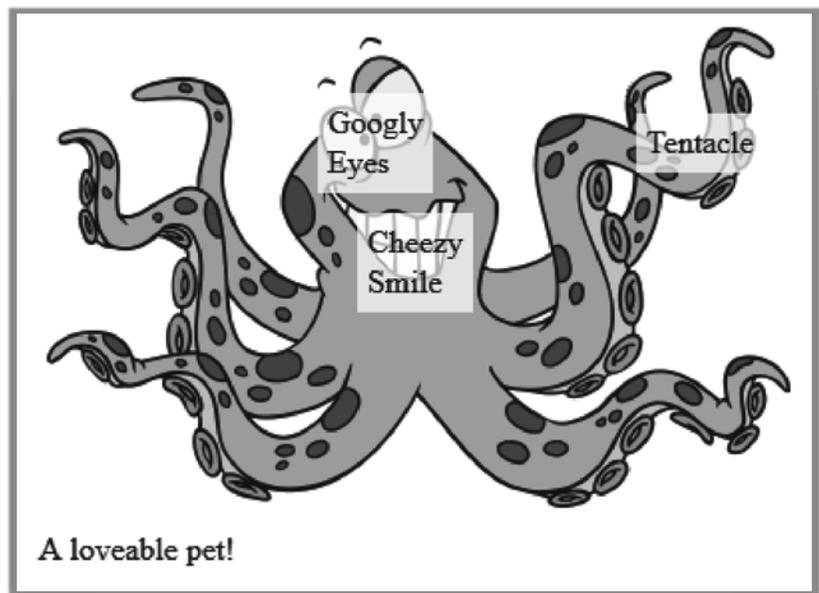
The style rule below will change the behavior of the individual list items `` in our annotated figure. Most importantly, we change the positioning to absolute, which will allow the top and left properties we set on each element to take effect. Those elements will now be positioned according to the pixel coordinates from the top-left corner of the parent figure. In addition, we add a **width**, **height**, some **padding** and a **background-color** to make the annotations stand out from the image.

```
.annotated li{
  position:absolute;
  width: 50px;
  height: 40px;
  padding: 5px;
  background-color:rgba(255,255,255,0.8);
}
```

The format of the **background-color** property one you have not seen before. You know that colors can be set using Red, Green, and Blue components (RGB), and each component is a value between 0 and 255. You can set those values using normal decimal numbers or hexadecimal numbers #00 - #FF. So why do we have a fourth value “0.8” at the end?

This fourth value is called the **alpha** or **transparency** value. The alpha value should be between 0.0 and 1.0. A value of 0.0 means the color is completely transparent and won't be seen at all. A value of 1.0 means the color is completely solid and won't let anything underneath show through. Any value in between such as 0.2, 0.5, or 0.8 will block 20%, 50%, or 80% of the underlying image.

Now we can see good results from our hard work. Each list item is positioned on top of the image with correct locations, and the text has a nice background that makes it readable but still shows some of the underlying image.



In the next lesson you will learn how to make annotations appear only when the user hovers a mouse on top of them. But first, it's time to put your new knowledge to work on the Aquamaniacs website.



Work with Me: Aquarium Annotations

In this activity you are going to add a new “Aquarium” page to your Aquamaniacs website. The aquarium will have an annotated image of fish swimming in the fish tank.

1. To begin, use Windows Explorer or Mac OS Finder to copy the “aquarium.html” file from your “Activity Starters/Chapter04” folder to your “MyProjects/Aquamaniacs” directory. Also copy the “aquarium.jpg” file from “Activity Starters/Chapter04/PagePhotos” to your “MyProjects/Aquamaniacs/PagePhotos” directory.
2. Load the “aquarium.html” file from your Aquamaniacs folder into your web browser and make sure you can see the picture and list of annotations underneath the caption.

You will not have to make any changes to “aquarium.html” for this activity. All of your work will be inside the “global.css” file.

Aquarium Tank

Spotting Seven Fish



A group of fish swimming in an aquarium.

- Fish 1
- Fish 2
- Fish 3
- Fish 4
- Fish 5
- Fish 6
- Fish 7

- Now, load your “Aquamaniacs/SiteStyle/global.css” file in your Komodo Edit software. Scroll down to the bottom of the page, and add the three styles for “annotated” figures, lists, and list items exactly as described in the lesson.

```
.annotated {  
    position:relative;  
}  
.annotated ul{  
    list-style:none;  
    position:absolute;  
    top:0;  
    left:0;  
    font-size:16px;  
}  
.annotated li{  
    position:absolute;  
    width: 50px;  
    height: 40px;  
    padding: 5px;  
    background-color:rgba(255,255,255,0.8);  
}
```

- Save your changes and reload “aquarium.html” in your web browser. You should now see the “Fish1” through “Fish7” annotations appear in different places on the image.

In the next lesson you’ll learn how to apply some other special effects to your figure annotations.

Spotting Seven Fish



A group of fish swimming in an aquarium.

SAMPLE SOLUTION GUIDE

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

Chapter Seven Activity (Grooving and Shaking)

In this activity your student is going to finish their new “Groovy Fish” and “Funny Fish 1” pages by adding some gradients and transformations on their own.

The student will need to add the following code (in bold text) to the “global.css” file:

```
#gradient1{
    background-image:url("fish.png"),linear-gradient(red, yellow);
    background-repeat: no-repeat;
    background-position: right bottom;
}

#gradient2{
    background-image:url("fish.png"),linear-gradient(to bottom right, purple, red,
                                                    yellow, green);
    background-repeat: no-repeat;
    background-position: right bottom;
}

#gradient3{
    background-image:url("fish.png"),linear-gradient(35deg, yellow 20%, green 20%,
                                                    green 50%, yellow 50%, yellow 80%, green 80%);
    background-repeat: no-repeat;
    background-position: right bottom;
}

#gradient4{
    background-image:url("fish.png"),radial-gradient(pink,blue);
    background-repeat: no-repeat;
    background-position: right bottom;
}
```

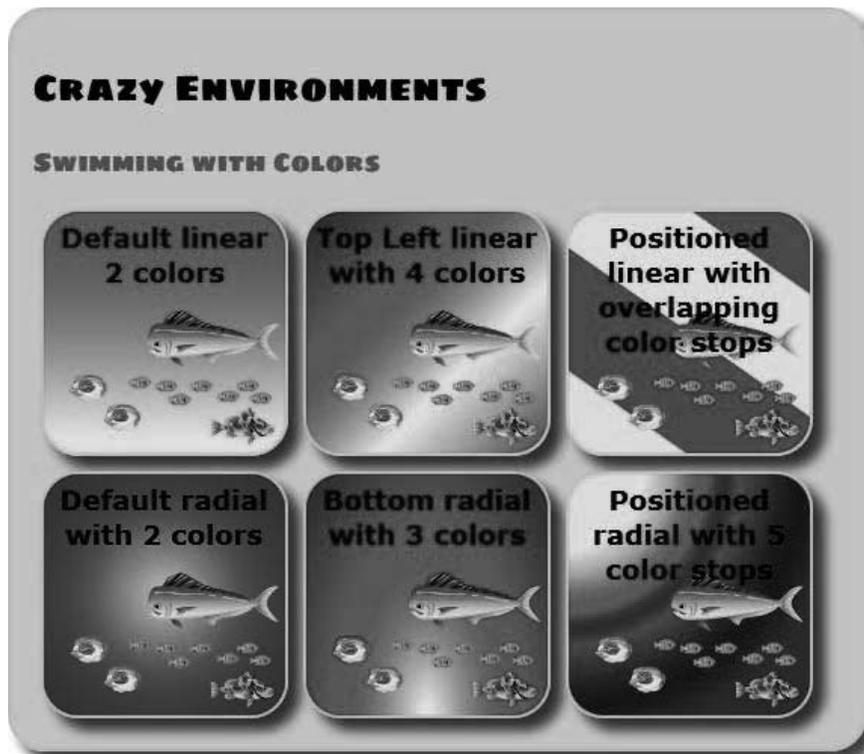
```
#gradient5{
    background-image:url("fish.png"),radial-gradient(at bottom, yellow, red,
                                                    purple);

    background-repeat: no-repeat;
    background-position: right bottom;
}

#gradient6{
    background-image:url("fish.png"),radial-gradient(at top left,
                                                    yellow 20%,orange 30%, red 40%, green 50%, purple 80%);

    background-repeat: no-repeat;
    background-position: right bottom;
}
```

When the student is done with these changes, the “groovyfish.html” file should look like the following when viewed in a web browser:

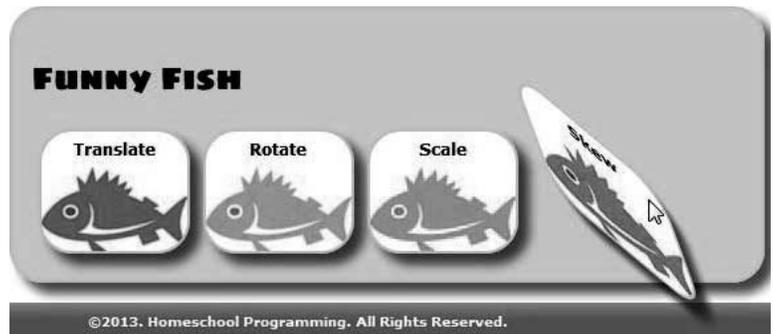
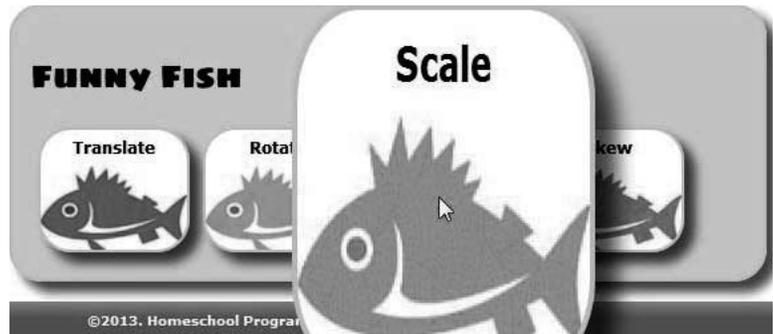


Chapter Seven Activity (Grooving and Shaking)

In the second part of the activity, the student will add the following code to the “global.css” file:

```
#fish3:hover
{
    transform: scale(2,3);
    -webkit-transform: scale(2,3);
}
#fish4:hover
{
    transform: skew(20deg,40deg);
    -webkit-transform: skew(20deg,40deg);
}
```

After the student has completed these changes, the “funnyfish1.html” file should be reloaded in a web browser to check the results. Each of the boxes should look like the ones shown below. Remember, you need to hover your mouse cursor over the image to get the transformation effect.



The completed HTML file for this activity is “Activity Solutions/Chapter07/funnyfish1-assignment.html” and “Activity Solutions/Chapter07/groovyfish-assignment.html” and the completed CSS file is “Activity Solutions/Chapter07/SiteStyle/global-assignment.css” located in the directory where you installed the “Solution Files”. You can also access the “Your Turn” solutions to see a live example of the web pages documented in this Solution Guide.