

Week 1

# Welcome to Introduction to Computer Programming + Game Development

Who wants to be a game developer ?

First we need to learn a thing or two about  
where our games are going to run;  
Desktop and Mobil devices over the web.

Lets start with some web basics.

## Hypertext Markup Language (HTML)

HTML (HyperText Markup Language) is the language used to create web page documents. There are a few versions of HTML in use today: HTML 4.01 is the most firmly established and the newer, more robust HTML5 is gaining steam and browser support. Both versions have a stricter implementation called XHTML (eXtensible HTML), which is essentially the same language with much stricter syntax rules. We'll get to the particulars of what

language with much stricter syntax rules. We'll get to the particulars of what makes the various versions different in [Chapter 10, What's Up, HTML5?](#)

HTML is not a programming language; it is a markup language, which means it is a system for identifying and describing the various components of a document such as headings, paragraphs, and lists. The markup indicates the document's underlying [structure](#) (you can think of it as a detailed, machine-readable outline). You don't need programming skills—only patience and common sense—to write HTML.

The best way to learn HTML is to write out some pages by hand, as we will be doing in the exercises in this book. If you end up working in web production, you'll live and breathe HTML. But even hobbyists will benefit from knowing what is going on under the hood. The good news is that it's simple to learn the basics.

## Cascading Style Sheets (CSS)

While HTML is used to describe the content in a web page, it is Cascading Style Sheets (CSS) that describe how that content should *look*. In the web design biz, the way the page looks is known as its [presentation](#). That means fonts, colors, background images, line spacing, page layout, and so on... all controlled with CSS. With the newest version (CSS3), you can even add

special effects and basic animation to your page.

CSS also provides methods for controlling how documents will be presented in contexts other than the traditional desktop browser, such as in print and on devices with small screen widths. It also has rules for specifying the non-visual presentation of documents, such as how they will sound when read by a screen reader (although those are not well supported).

Style sheets are also a great tool for automating production because you can change the way an element looks across all the pages in your site by editing a single style sheet document. Style sheets are supported to some degree by all modern browsers.

Although it is possible to publish web pages using HTML alone, you'll probably want to take on style sheets so you're not stuck with the browser's default styles. If you're looking into designing websites professionally, proficiency at style sheets is mandatory.

## How the web works:

The [Internet](#) is a network of connected computers. No company owns the Internet; it is a cooperative effort governed by a system of standards and rules. The purpose of connecting

computers together, of course, is to share information. There are many ways information can be passed between computers, including email, file transfer (FTP), and many more specialized modes upon which the Internet is built. These standardized methods for transferring data or documents over a network are known as [protocols](#).

The [Web](#) (originally called the World Wide Web, thus the “www” in site addresses) is just one of the ways information can be shared over the Internet. It is unique in that it allows documents to be linked to one another using [hypertext](#) links—thus forming a huge “web” of connected information. The Web uses a protocol called [HTTP \(HyperText Transfer Protocol\)](#). That acronym should look familiar because it is the first four letters of nearly all website addresses

# Browser

# Server

1 Type in a URL or click on a link in the browser.

http://www.jens.kitchensite.com

2 The browser sends an HTTP request.

HTTP request

## Server Contents



3 The server looks for the file and responds with an HTTP response.

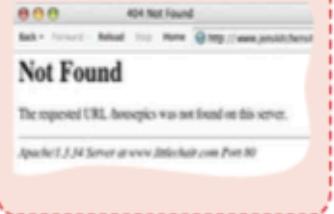
HTTP response

index.html

"I see that you requested a directory, so I'm sending you the default file, index.html. Here you go."

Oops, no file

If the file is not on the server, it returns an error message.



4 The browser parses the document. If it has images, style sheets, and scripts, the browser contacts the server again for each resource.

kitchen.gif  
spoon.gif  
kitchen.css

5 The page is assembled in the browser window.



Our Main Tool: during this class will be JavaScript.

We will (only) overview - HTML & CSS; as these are simple (containers) for our main TOOS of JavaScript.

So What is JavaScript?

Have you ever heard of JavaScript?

Wiki -JavaScript

JavaScript ([/ˈdʒɑːvə skɪpt/](#)),<sup>[7]</sup> often abbreviated as **JS**, is a high-level, interpreted

programming language. It is a language which is also characterized as [dynamic](#), [weakly typed](#), [prototype-based](#) and [multi-paradigm](#).

Alongside [HTML](#) and [CSS](#), JavaScript is one of the three core technologies of the [World Wide Web](#).<sup>[8]</sup> JavaScript enables interactive [web pages](#) and thus is an essential part of [web applications](#). The vast majority of [websites](#) use it,<sup>[9]</sup> and all major [web browsers](#) have a dedicated [JavaScript engine](#) to execute it.

As a multi-paradigm language, JavaScript supports [event-driven](#), [functional](#), and [imperative](#) (including [object-oriented](#) and [prototype-based](#)) [programming styles](#). It has an [API](#) for working with text, [arrays](#), dates, [regular expressions](#), and basic manipulation of the [DOM](#), but the language itself does not include any [I/O](#), such as networking, storage, or graphics facilities, relying for these upon the host environment in which it is embedded.

Initially only implemented [client-side](#) in web browsers, JavaScript engines are now embedded in many other types of host software, including [server-side](#) in web servers and databases, and in non-web programs such as word processors and [PDF](#) software, and in runtime environments that make JavaScript available for writing mobile and desktop applications, including desktop widgets.

Although there are strong outward similarities between JavaScript and [Java](#), including language name, [syntax](#), and respective [standard libraries](#), the two languages are distinct and differ greatly in design; JavaScript was influenced by programming languages such as [Self](#) and [Scheme](#).<sup>[10]</sup>

---

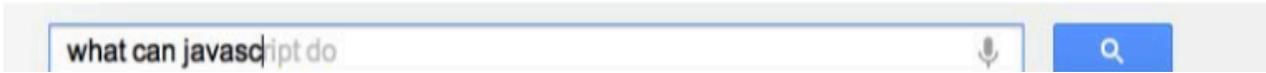
JavaScript is a lightweight but incredibly powerful scripting language. We most frequently encounter it through our browsers, but JavaScript has snuck into everything from native applications to PDFs to ebooks. Even web servers themselves can be powered by JavaScript.

As a [dynamic programming language](#), JavaScript doesn't need to be run through any form of compiler that interprets our human-readable code into something the browser can understand. The browser effectively reads the code the same way we do and interprets it on the fly.

JavaScript is also [loosely typed](#). All this means is that we don't necessarily have to tell JavaScript what a variable is. If we're setting a variable to a value of 5, we don't have to programmatically specify that variable as a number. As you may have noted, 5 is already a number, and JavaScript recognizes it as such.

In short, JavaScript allows you to create highly responsive interfaces that improve the user experience and provide dynamic functionality, without waiting for the server to load up a new page. For example, we can use JavaScript to do any of the following:

- Suggest the complete term a user might be entering in a search box as he types. You can see this in action on Google.com ([Figure 19-2](#)).



what can javascript do





what can javascript do  
what can javascript be used for  
what can javascript do for a website  
what can javascript programs do

*Figure 19-2. Google.com uses JavaScript to automatically complete a search term as it is typed in.*

JavaScript - will be our <main> tool for game development...

HTML & CSS are only - like a tool box that holds your tools...

Your instructor, ***Coach Arthur*** will help you, and guide you through what you need... Having exposure to these will help your understanding of what is going on.

For example: if you want to teach some one how to multiply, giving them

an overview of addition is helpful for the young person learning.

## Adding JavaScript to a Page

Like CSS, you can embed a script right in a document or keep it in an external file and link it to the page. Both methods use the **script** element.

### Embedded script

To embed a script on a page, just add the code as the content of a **script** element:

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

### External scripts

The other method uses the **src** attribute to point to a script file (with a *.js* suffix) by its URL. In this case, the script element has no content.

```
<script src="my_script.js"></script>
```

The advantage to external scripts is that you can apply the same script to multiple pages (the same benefit external style sheets offer). The downside,

## Comments

JavaScript allows you to leave comments that will be ignored at the time the script is executed, so you can leave reminders and explanations throughout your code. This is especially helpful if this code is likely to be edited by another developer in the future.

There are two methods of using comments. For single-line comments, use two slash characters (`//`) at the beginning of the line. You can put single-line comments on the same line as a statement, as long as it comes after the statement. It does not need to be closed, as a line break effectively closes it.

```
// This is a single-line comment.
```

Multiple-line comments use the same syntax that you've seen in CSS. Everything within the `/* */` characters is ignored by the browser. You can use it to “comment out” notes and even chunks of the script when troubleshooting.

```
/* This is a multi-line comment.  
Anything between these sets of characters will be  
completely ignored when the script is executed.  
This form of comment needs to be closed. */
```

# JavaScript Variables

A variable is like an information container. You give it a name and then assign it a value, which can be a number, text string, an element in the DOM, or a function—anything, really. This gives us a convenient way to reference that value later by name. The value itself can be modified and reassigned in whatever way our scripts' logic dictates.

The following declaration creates a variable with the name “foo” and assigns it the value 5:

```
var foo = 5;
```

There are 3 type of variables:

## 1. Numeric

```
var foo = 5;
```

## 2. String

```
var test = "This is a test";
```

## 3. Boolean

```
var flag = true;
```

All statements end with a semicolon";"

The "=" is assignment operator

JavaScript variable are case sensitive.  
Therefore the variable flag is different than the variable Flag.