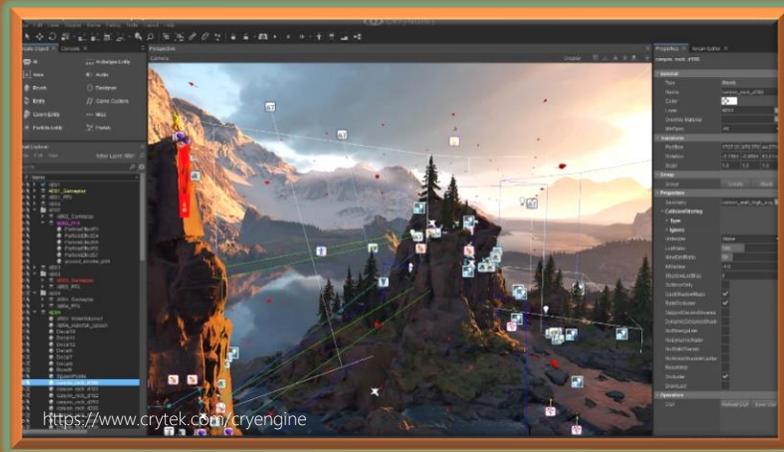




# Game Engine [Game Development & Game Design]



A Game Engine is, to sum it up in an understandable way, is a suite of tools that give the user the ability to complete otherwise inoperable tasks, as it relates to video games of course. For example, Unreal Engine 3 was comprised of a bunch of third party tools to do each different thing such as level editing, code editing, animation, rigging, etc, resulting in the software

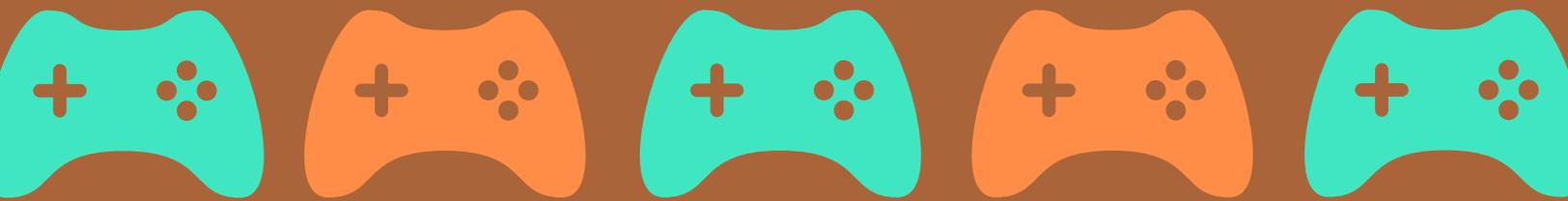
being very expensive, and very counter-intuitive comparatively. The release of Unreal Engine 4 brought with it a complete set of tools developed and integrated by Epic Games.<sup>[2]</sup> (Epic Games: What is Unreal Engine 4) This allowed them to make the software free and allowed for a much more productive and convenient workflow. Game Engines are the core of a project within the gaming world because it is where each individual arm of the contributors is put together and functions how it was



initially intended, without the engine it would not be possible to put each of these pieces together. Another big name in the engine industry is Crytek's CryEngine. The CryEngine is a competing tool used widely throughout the industry, and engine is built into the name of the engine! If you scroll through their website, they even break down what an engine is and how it works to fill people in on why they should pick it over, say, Unity.

See their website and more here! Interested in Unity or comparing how they break their engines down? Check out Unity's website too! Most of the leading engine's carry their own Question and Answers websites like Unity Answers, UE4 AnsweHub, and CryEngine Answers and Ideas Platform.



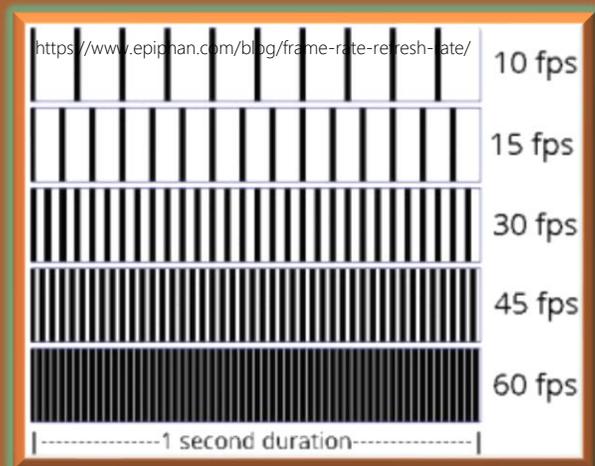


## Framerate [Game Development & Game Design]

Framerate, also known as FPS by gamers and game developers, is the quality of the speed of playback. That is an odd definition, but what I mean is that the framerate, being the number of images shown every second to make the moving picture you see (ex. Video, flipbook), changes how good the video looks during playback by enhancing the same quality images. When I say quality of speed, it just means it shows more per second rather than showing more pixels per inch. [Epiphan Video](#), the makers of extremely high-end video/audio production, conversion, streaming, hosting, and recording software, and even all-in-one streaming hubs, go in depth into the values, whys, and hows of framerate and



compare it to refresh rate in a blog post! Writer, [Adam Frame](#) goes into detail about how in the industry, big names like Peter Jackson are using technology to ensure maximum quality and framerate whilst not having to sacrifice file size; they use their own encoding and compression technology to elaborate on the need of certain tools to stream and record in high definition without having to spend the "356 megabytes per second,"<sup>[1]</sup> rather you can spend a mere "1.5 megabytes per second."<sup>[1]</sup> Also, when you watch videos on internet, if you are like us with Xfinity and have a limited amount of data usage on your router plan per month, it uses data; going from 30 frames per second to 60 frames per second can double the size of the file, in turn doubling the internet usage and cutting your time using the internet in half. With all of the streamers on: [Twitch.TV](#), [YouTube](#), [Facebook](#), and similar sites, and the video streaming services like: [Netflix](#), [Hulu](#), [HBO Now](#), [Xfinity](#),



Soon to be [Disney Play](#), [SHOWTIME](#), [Dish Anywhere](#), and more similar apps, and especially with all of the audio streaming apps with the advances in sound technology lately: [Google Play Music](#), [Apple Music](#), [Spotify](#), [Pandora](#), [Groove](#), [Amazon Prime](#), [TIDAL](#), and more, there are a lot of ways to max out that data cap today; this applies double on phones with mobile providers. Framerates are imperative across all industries whether you are optimizing the next AAA video game IP or animating the next big Disney property!



## Backend [Game Development & Mobile Development]

Backend development is the core of most software, apps, and animation, video more!

Backend is a blanket term for the development that user will not see, but interact with the Long of Team where you can learn

```
252 document.getElementById('updatePhotoDescription') {
253 }
254
255 function updatePhotoDescription() {
256   if (descriptions.length > (page * 10) + (currentImage.substring(0, 10))) {
257     document.getElementById('updatePhotoDescription') {
258   }
259 }
260
261 function updateAllImages() {
262   var i = 1;
263   while (i < 10) {
264     var elementId = 'foto' + i;
265     var elementIdBig = 'bigimage' + i;
266     if (page * 10 + i - 1 < photos.length) {
267       document.getElementById(elementId).src = 'images/' + page + 'photos/' + i + '.jpg';
268       document.getElementById(elementIdBig).src = 'images/' + page + 'photos/' + i + '.jpg';
269     }
270   }
}
```

nt is the core of most even videos, movies, games, cars, and development is a period or part of consists of what the rather how they can environment. Josh Treehouse, a website to code and learn

about those in the industry with articles and blog posts, explains back-end development as, “a server, an application, and a database,” and continues, “We call a person that builds all of this technology to work together a backend developer.”<sup>[3]</sup> He also goes on to examine front-end versus backend development, which we will get into a tad bit later with another term! In the video game industry, the backend developers, like me, are the ones who implement the code for the project, built the engine, and work on the UX. An artist may give me a set of user interface elements, for example, and elaborate to me how they wish for it to look and act in game; my job would be to implement all these items how the artist intends and make sure they function fully and properly. When you move your mouse to look around, click or pull to shoot, use WASD or sticks to move around, use all the controls and functions, this is where backend development shines. Playing a game is entirely dependent on the integration of the front-end elements with the backend elements, because the backend ones are what make the others function and work, and without having a good backend framework, your entire system could crumble under the weight of the mess you may have created. Backend management is a lot like time management and has a lot of time management as a part of it. Backend uses the game engine to program all this functionality and integrate it.





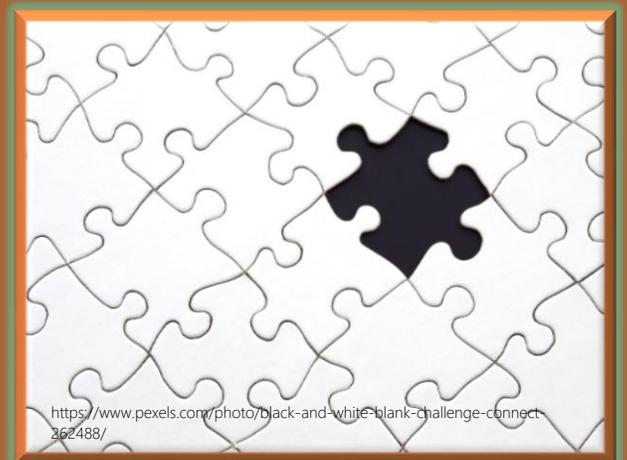


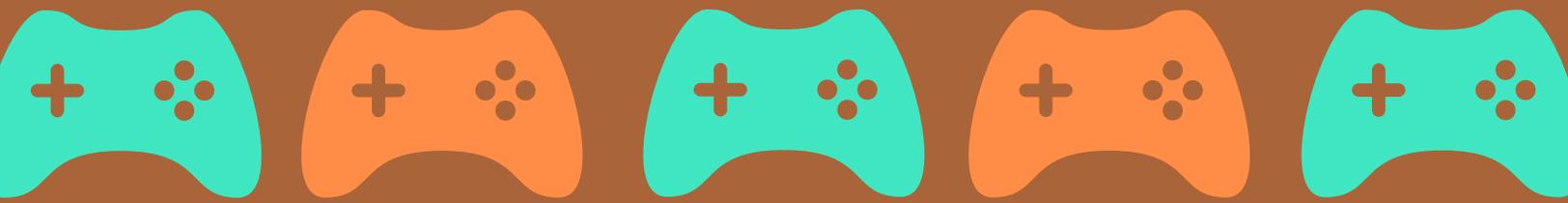
## Bugs [Game Development & Software Development]

Bugs are unwanted functionality in code or builds. What that means is this: you are building a program, and now you are trying to get a button to open a document; when you compile and run, the button, instead of opening the document, closes the application. This is a bug, because it is something that was programmed, but was not the intended function of the item or set. Bugs can appear from the start, or they can worm their way in while the player saves data over time to slowly destroy a save game file. Besides game ruining bugs, you can even have simple ones, say when you jump, you get higher with each leap because the



code makes the velocity higher each time rather than setting it or vice versa. One notable example that is at the top of my head still is The Witcher 3. The game is most likely the best game ever made, but it is riddled with bugs everywhere you turn because of the scale and complexity. One example is when swimming; in small or medium areas when you are looting, you can get permanently caught on nothing and suffocate under water<sup>[5]</sup>. Therefore, there are bug testers and quality control with video games and software (video games are inherently software, but I mean computer and mobile applications for software), if they did not have the QC, you would inevitably have infinitely more of these bugs. Just as in real life, when one bug comes in, many can start eating from your pantry; eventually you won't have enough in your pantry and your computer will crash. This is not the case every time, but it is easy to compare even the rarity to real bugs; the odds are you have many bugs in your house right now, but only seldom do they really become so bad you are required to spend money on them. People don't like creepy crawly bugs for many of the same reasons people don't like the bugs in computer fields, and it is not too hard to see why. Another bug that is very prominent in the gaming community is the famed Swing Set Launcher in Grand Theft Auto IV<sup>[6]</sup>, all you must do is gas repeatedly on the edge of a swing set, the off you go into the sky, a million miles an hour! That glitch has most likely fallen victim to one of many physics glitches. Physics are one of the toughest things to nail without bugs in games because of the complexity of the code behind it. Most companies have **dedicated websites to bug reporting!**





## Rendering [Game Development & Game Art]

Rendering is the process where the computer takes the input (game data, animation data, etc...) and treats it as a coloring book, then goes over it and shades it. It may sound dumbed down, but it really isn't, rendering is when the computer takes a two-dimensional outline of something, then goes over it with color and shading to make it appear three-dimensional. This process is where your graphics card comes in handy. Every video, almost since the beginning of computer and console

games, has been using rendering in some form; that is why video games single handedly pushed graphics cards to the levels they are at today. Without DOOM (1993), Mortal Kombat (1992), Unreal Tournament (1999), Castle Wolfenstein (1981), Wolfenstein 3D (1992), Quake (1996), and other games like them in that defining era, we would not have the computers, technology, medical equipment, cars, industry, and more that we cherish today. Rendering is making so many breakthroughs today, that Epic Games, makers of Unreal Engine, have released Unreal Studio for companies looking to produce high quality interactive augmented reality, virtual reality, and mixed reality experiences. They (Epic Games) are now working with Volkswagen, BMW, Animech, and NVIDIA on these experiences all over the world<sup>[7]</sup>! Not only is it being used to please clients of car and manufacturing companies, but also helping render genomes, cells, and other scientific items in a new separate world that allows them to research at speeds and in ways that were never possible by hand. Many of the car maker's commercials are done almost purely in CGI and VFX; [take this video](#) for example. This is Audi's vision of the future, and they haven't created all the tech yet, so they were forced to animate and render much of the video and use Visual Effects to spice up the rest of the footage they did record. Any movie you watch with CGI, it has been rendered. Rendering breaks many walls and barriers that lead to a better future for every industry on the face of this planet.





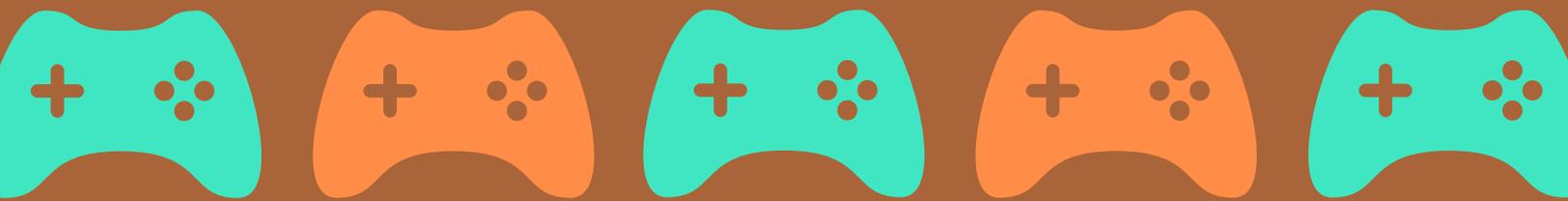
## Format [Game Development and Graphic Design]

There are many meanings and uses of the word format in the career fields that I am covering, but the one that most closely relates to my Game Development career path is the file format. File formats can change many things from the simplest, like if an image is meant for Windows, Linux, Mac, or all, to the complicated, like if a game is for Xbox, PlayStation, or Switch. When importing audio into Unreal Engine, for example, a .wav file type is for sounds easily playable and repeatable like gunshots, footsteps, callouts, etc, while an .mp3 and others are used for everything from dialog and subtitles to arrays of sounds. In fact,



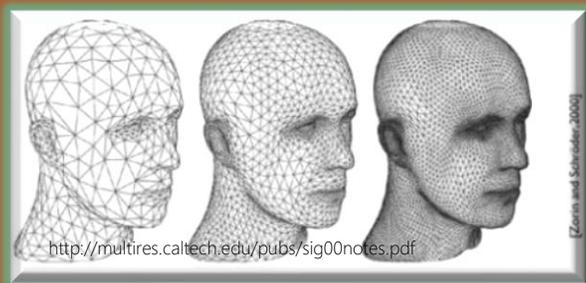
there are many forums where people ask questions and discuss the results with experts. One of these that I was browsing earlier today by username The Incredible Smoker gave me insight into the relation between animation and file formats and the depth they go into with certain file formats<sup>[8]</sup>. Epic Games made a tutorial for just about every feature in the engine, and in their [audio tutorial](#), it shows a lot of the specifics with different audio formats like which contain stereo: front, back, and side audio, and which are mono. Even with video, animation, game development, and architecture have file formats, and they can range from a 3D CAD design for the next largest sky scraper to a level containing all the assets used to make up the game. When you are a graphic designer you work with clients on their needs, which can include the need for a different format so they can use the images or animations in certain programs or software. They may even need a specific format to ensure that it will play on their computers, because they are importing it into Unreal Engine instead of Unity. When you make a game with Unreal Engine, the file format of programmed assets is .UASSET, which is only usable in the engine itself and will not work in another engine, program, or application. The reason for this is because the Unreal Engine has its own set of unique code for games in it to work, and this file format of .UASSET is easily readable and writable by the engine, whereas a non-custom file type could be extremely counter-productive.





## Polygons [Game Art]

Polygons are the backbone of 3D models, literally. A 3D model is made up of sometimes hundreds of thousands of polygons, which are tiny multi-sided two-dimensional shapes that connect to make a three-dimensional object. Polygons are used in many fields for many things, and even Autodesk, one of the premier



3D modeling sweets in the industry, develops its tools to ensure that all fields have what they need to do their jobs. Everything from film and video games, to the internet [10]! Polygons are generated by the computer to be used in a 3D space, which is why 2D games use sprites instead of models, because they do not need the polygons, and saving the processing power to show many sprites on screen is much more beneficial and optimized. Polygons can be in the Game

Development field, but not very much, only the optimizing part; this means we never directly deal with them physically, only in the code parts. Polygons are the basis of 3D gaming and the core reason for game engines and 3D gaming now-a-days. Almost every field in the video games industry will have a direct or indirect interaction with polygons, or something they effect within their job.

## Modeling [Game Art]

3D Modeling is a very long and complex process for those who do not do it as a career, and even with a career in it, you may take 12 hours to do an extremely intricate and detailed piece depending on what you are using it for. Even the professors at California Technical Institute go over many subjects within Modeling, but a few stood out to me. The idea of subdivision where a surface can be divided up

into more Polygons (see above) and contain more small surfaces for finer details, but also sacrifices memory [9]; this fascinated me because it relates to using commands, functions, queries, etc in programming, which also uses more memory. Piecewise Polynomial Curves are another idea I found interesting because they are basically splines, only it more refers to the equation by which a modeler would be able to customize the curve of the spline they are making. This would give better detail to the mesh or model that is being crafted and give the crafter more control over the shape, quality, size, and dimensions of their piece. [9] The math seems similar to the Game Development program, but beyond that, a modeler has much more physical artistic skill and much stronger intuition for design and color schemes.



into more Polygons (see above) and contain more small surfaces for finer details, but also sacrifices memory [9]; this fascinated me because it relates to using commands, functions, queries, etc in programming, which also uses more memory. Piecewise Polynomial Curves are another idea I found interesting because they are basically splines, only it more refers to the equation by which a modeler would be able to customize the curve of the spline they are making. This would give better detail to the mesh or model that is being crafted and give the crafter more control over the shape, quality, size, and dimensions of their piece. [9] The math seems similar to the Game Development program, but beyond that, a modeler has much more physical artistic skill and much stronger intuition for design and color schemes.



## Animation [Computer Animation]

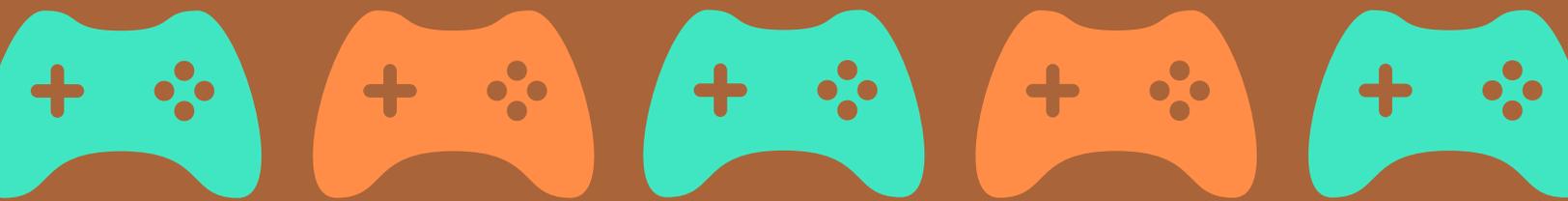
Animation is the step after modeling and rigging, but before actually putting them in the game to start programming gameplay. Animation is adding emotion, character, and personality to an object; it is adding movement, movement type, skills, jumps, and everything that the end-user will see physically in the game world.<sup>[11]</sup> Andy Serkis recently worked together with Epic Games to bring his newest game to life in ways never seen for Planet of the Apes. They focused much of their work on facial animation because they knew that it is the heart of character emotion and having static facial animations can lead to an unbelievable and disconnecting experience like Dead Island (again, great game, just had its issues like every game does). There are those who can animate entire cutscenes single handedly and add zany personal and requested touches to it, and there are others who professionalize with motion capture software and hardware. There are many uses and ways to do animation with all the technology that is available to even indie developers because of modern sciences and mathematics. James Cameron is using a new home-made setup for motion capture under water and it takes thousands of little cameras to get the latency of the water right while capturing the actors in the water.<sup>[12]</sup>



## Baking [Game Art]

Baking is the process where a computer processes some information on compile or run, rather than doing it real-time while the player is in game. This is done to save processing power and speed up the game. A very popular use of this in games and animation is the lighting<sup>[12]</sup>. You can pre-visualize the lighting in a game with a lightmap so that the shadows can be rendered real time when it counts but save on the lighting that isn't as important to render real time and can easily be done with the same purpose before play. This helps especially in large scale projects in games like Grand Theft Auto, Horizon Zero Dawn, Mirrors Edge, Mass Effect, BioShock, and so many more. This also gives developers the opportunity to add more lighting and more detail into the scenes they create, putting less limitations on the creator and giving them freedom to show their skill and their style. Baking is one of the many things that has advanced gaming and animation to the point we can have The Witcher 3 sized worlds with the graphics to back such a marvel. As gaming advances, I have a suspicion that lighting will all be done real time, but for now, we can't process that much information at once. Lighting is done per particle, so baking it out is an exponentially more optimized solution.





## Psychographics [Graphic Design]

Polygons Psychographics are, to sum it up, the statistics that you get when you do studies of groups of people to find their interests or personal preferences for certain things. A few examples include finding what kind of interface design people prefer, or what kinds of levels will play and in what ways based on how people play and how to change the less favorable areas, how to (and I am completely



against this one and strongly dislike the greedy people of the industry) implement microtransactions into a game and get the peak value out of the system with the player base.<sup>[13]</sup> The uses of psychographics can even go as deep as what will make an audience connect with a certain character or event that happens in a game, such as Maria's death scene in Gears of War 3 when Dom broke into tears. Even controls can play into the psychographics, and some games have changed controls based on the

psychographics of the first game for the sequel. Many games have built in trackers to give the developers feedback on how the masses are playing and what parts of the game they are enjoying. This gives developers a grasp on how people see the game, and in turn how they see the company and industry. One example, by opinion of course, is that Guitar Hero III was the best in the entire franchise.

## Front End [Web Design]

I want to start by giving a reference to an incredible article from [Dain Miller, Presidential Innovation Fellow at The White House, and mentor for developers at starthere.fm](#), that breaks down most of the essentials for user interface versus user experience, and the cores of both. Of course, with good interface you have great experiences to work from<sup>[14]</sup>. The front end is not just about what the end user sees, like I've said it is

also about the experiences, so you must dig deeper into the whole project to understand what ingredients make up a truly immaculate product. The Front End is about how the end-user interacts with the supplied content. In video games, the Front End is about how the player uses and interacts with the game world, it is about the

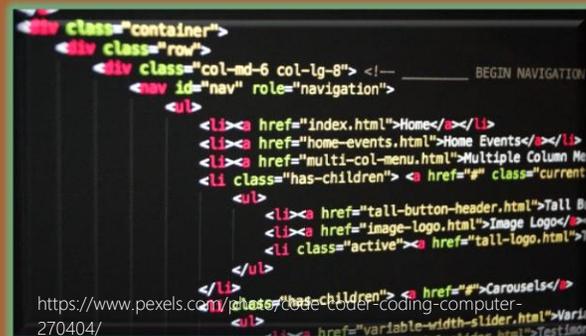


feedback that is given that the player can see, hear, or otherwise understand in some physical capacity. Every person in every industry is doing some form of Front End work at some point. For example, cars are more back end, but the Front End (interior and looks) break the deal.

# Markup [Web Design]

Markup are the development languages such as HTML, CSS, JavaScript, jQuery, SQL, Python, Ruby, and so many more, that are meant for defining, processing, and displaying text. Here is an example, in HTML, putting text on a page is as simple as `<p>This is some text!</p>`. That is Markup in HTML, how I wrote the code, and I can even write `<script>document.write(This is some text!);</script>` and it will write some text to the screen in JavaScript!

design websites at their cores (which is primarily text Showing text is critical to the gaming and media where they are. Imagine Snapchat with no text or text text or a Microsoft Word without text or a phone keyboard without text would



```
<div class="container">
  <div class="row">
    <div class="col-md-6 col-lg-8"> <!-- BEGIN NAVIGATION
    <nav id="nav" role="navigation">
      <ul>
        <li><a href="index.html">Home</a></li>
        <li><a href="home-events.html">Home Events</a></li>
        <li><a href="multi-col-menu.html">Multiple Column Men
        <li class="has-children"><a href="#" class="current">
          <ul>
            <li><a href="tall-button-header.html">Tall But
            <li><a href="image-logo.html">Image Logo</a></li>
            <li class="active"><a href="tall-logo.html">Ta
          </ul>
        </li>
      </ul>
    </div>
  </div>
</div>
```

<https://www.pexels.com/photo/code-editor-coding-computer-270404/>

Markup is the code used to and show all the content content) to the end-user. programming, and without it, industries would not be Facebook without any text, in images, a Google without without text, a car radio without text, a digital serve no purpose. Without

being able to translate text to code, we could not communicate with computers at the most basic of levels, and that would make the advance libraries like C++ obsolete. Markup is where it all started, it is the granddad of code. Two great places to get started with markup, or even learn a little more, are [W3.org](http://W3.org), who officially maintain and create standards for various languages of code, HTML being prominent, and [Mozilla.org](http://Mozilla.org), the official developers, inventors, and keepers of JavaScript (don't confuse it with Java from Oracle). Both links will go in-depth on not only how to learn to code, but also the basics and core fundamental concepts of the languages, along with what their purpose and identity as a Markup Language is, and what it means.

