
Welcome to Foundations of Web Development!

Aim Code School
Vanessa Kasun

Allow me introduce myself...

- My name is Vanessa Kasun
 - I am a mother of a four year old girl named Bexlee
 - I have two dogs. A german shepherd named Max and a toy poodle named Caesar.
 - I graduated with a degree in Computer Science in 2019
 - I created a website for the North Omaha BID (freelance)
 - I created a political blogging website for politician Paul Anderson (freelance)
 - Started teaching for Aim Code School in January 2020
 - Instructing
 - Updating all course curriculum
-

-
- I enjoy yoga
Being outdoors
Web Development
Science
Humans
Learning
Reading
EVERYTHING!!

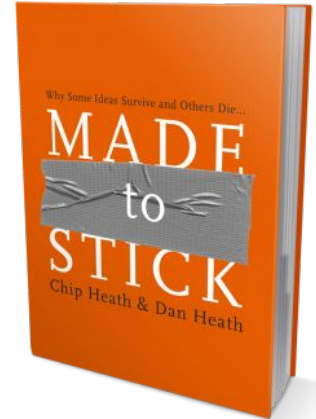


Continued...



First...What is a Developer?

1. A web developer is a programmer who specializes in, or is specifically engaged in, the development of World Wide Web applications using a client–server model.
2. Applications Developers Use
 - a. HTML
 - b. CSS
 - c. JavaScript
 - i. PHP
 - ii. ASP.NET
 - iii. Python
 - iv. Java





Different Developers

Front End Developers

Front-end Developers are responsible for implementing visual elements that users see and interact within a web application. In general, they are supported by back-end web developers, who are responsible for server-side application logic and integration of the work front-end developers do.

Back End Developers

Back-end developers are usually responsible for writing the web services and APIs used by front-end developers and mobile application developers. A back-end web developer is responsible for server-side web application logic as well as the integration of the front-end part.

FullStack Developers

Full Stack Developers are computer programmers who are proficient in both front and back end coding. Their primary responsibilities include designing user interactions on websites, developing servers and databases for website functionality and coding for mobile platforms.

Front-End Web Development Rules!



Tip

Plan your projects!

INVEST YOUR SKILLS

Be curious

Practice, Practice,
Practice!



Front-End Languages

- HTML/CSS
 - JavaScript
 - jQuery/ Bootstrap
 - CSS Preprocessors
 - SCSS
 - SaSS
 - RESTful Services/APIs
 - Vue
 - React
 - Angular
 - PHP
 - MySQL
-



Front-End Tools

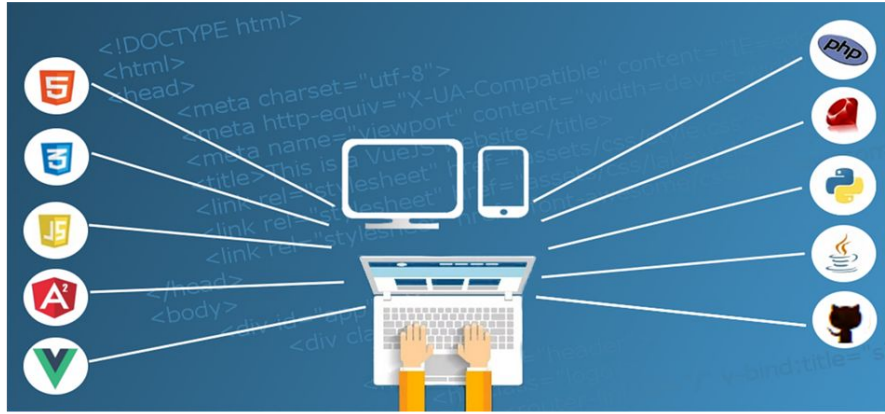
- PhotoShop / Adobe Illustrator
 - IDE
 - Visual Studio Code
 - IntelliJ
 - Atom
 - Boilerplates
 - CMS (content management systems)
 - Git/Version Control
 - Web Hosts
 - Google Fonts
 - Servers/Hosts
-

This is an example of what VERY BASIC front-end web development CODE looks like... HTML....CSS....JS

```
HTML
26+ <!-- /end gameboard -->
27+ <!-- guide -->
28+ <section class="guide clearfix">
29+   <h2> What is this? </h2>
30+   <p> Although coded entirely from scratch, this game is a (lackluster) copy
    of Gabriele Cirulli's 2048, http://2048game.com/.
31+ </p>
32+ </section>
33+ <section class="guide clearfix">
34+   <h2> How do I play? </h2>
35+   <p> Tiles with matching number values can be merged into a single tile,
    which receives the values' sum.
36+ </p>
37+   <p> To move the board, use the directional arrows - or swipe.</p>
38+   <div class="guide_arrow-wrap">
39+     <span class="guide_arrow"> &uHar; </span>
40+     <span class="guide_arrow"> &lHar; </span>
41+     <span class="guide_arrow"> &rHar; </span>
42+     <span class="guide_arrow"> &dHar; </span>
43+   </div>
44+   <p> To win, get a 2048 tile.
45+ </p>
46+ </section>
47+ <!-- /end guide -->
48+ </main>
49+ <!-- /end main -->
50+
51+ <!-- templates -->
52+ <script type="text/html" id="template_grid_cell">
53+   <div class="grid_cell"></div>
54+ </script>
55+
56+ <script type="text/html" id="template_tile">
57+   <div class="tile">
58+     <span class="tile_number"> </span>
59+   </div>
60+ </script>
61+ <!-- /end templates -->

CSS (SCSS)
1+ /*
2+   Variables:
3+   */
4+
5+ $grid-max-width: 500px;
6+ $grid-padding: 8px;
7+ $grid-border-radius: 5px;
8+ // white
9+ $color-background: #f2feea;
10+ $color-accent1: #f9d49a;
11+ $color-accent2: #dda8cf;
12+ $color-list: #00d0a4, #dd7373, #7d53de, #6622cc,
    #00bfb2, #c06ff2, #340068,
13+ #3e92cc, #d8315b, #1c0b19, #1c0b19;
14+ /**/
15+
16+ *,
17+ *:before,
18+ *:after {
19+   box-sizing: border-box;
20+ }
21+
22+ button,
23+ a {
24+   &:hover {
25+     cursor: pointer;
26+   }
27+ }
28+
29+ .clearfix::after {
30+   content: "";
31+   display: block;
32+   clear: both;
33+ }
34+
35+ html {
36+   min-height: 100%;
37+   width: 100%;
38+   font-size: 16px;

JS (Babel)
499+
500+ // if UP: check next position
501+ if (direction === "up") {
502+   getNext = this.y > 0 ? this.game.board[this.x][this.y - 1] :
    false;
503+   nextPositionArray.push(this.x, this.y - 1);
504+ } else if (direction === "right") {
505+   // if RIGHT: check next position
506+   getNext = this.x < 3 ? this.game.board[this.x + 1][this.y] :
    false;
507+   nextPositionArray.push(this.x + 1, this.y);
508+ } else if (direction === "down") {
509+   // if DOWN: check next position
510+   getNext = this.y < 3 ? this.game.board[this.x][this.y + 1] :
    false;
511+   nextPositionArray.push(this.x, this.y + 1);
512+ } else if (direction === "left") {
513+   // if LEFT: check next position
514+   getNext = this.x > 0 ? this.game.board[this.x - 1][this.y] :
    false;
515+   nextPositionArray.push(this.x - 1, this.y);
516+ }
517+ // Check if next position contains match or is empty
518+ isNextMatch =
519+   getNext &&
520+   getNext.tilesArray.length === 1 &&
521+   getNext.tilesArray[0].valueProp === this.valueProp;
522+ isNextEmpty = getNext && getNext.tilesArray.length === 0;
523+ //
524+
525+ // "check only" mode; only to check if tile can move
526+ if (checkFlag) {
527+   return isNextEmpty || isNextMatch ? true : false;
528+ } else if (isNextEmpty || isNextMatch) {
529+   // not "check only" mode; will actually run move logic
530+   this.setPosition(nextPositionArray[0], nextPositionArray[1]);
531+   this.removeOldPosition(getX, getY);
532+   // do NOT continue to move if a tile has matched - and
    therefore MERGED into adjoining tile
533+ }
```



Tip

Everything is connected. To become a good programmer you must understand all working parts of web development.

Other Information you need to know as a Front- End web dev

- Internet/Web Browsers
- API's
- DNS (domain name system)
- HTTP/Networks
- Web Hosting
- Command Line (CLI)
- Node.js
- JSON(JavaScript Object Notation)
- App Frameworks
- Back- End Languages and API

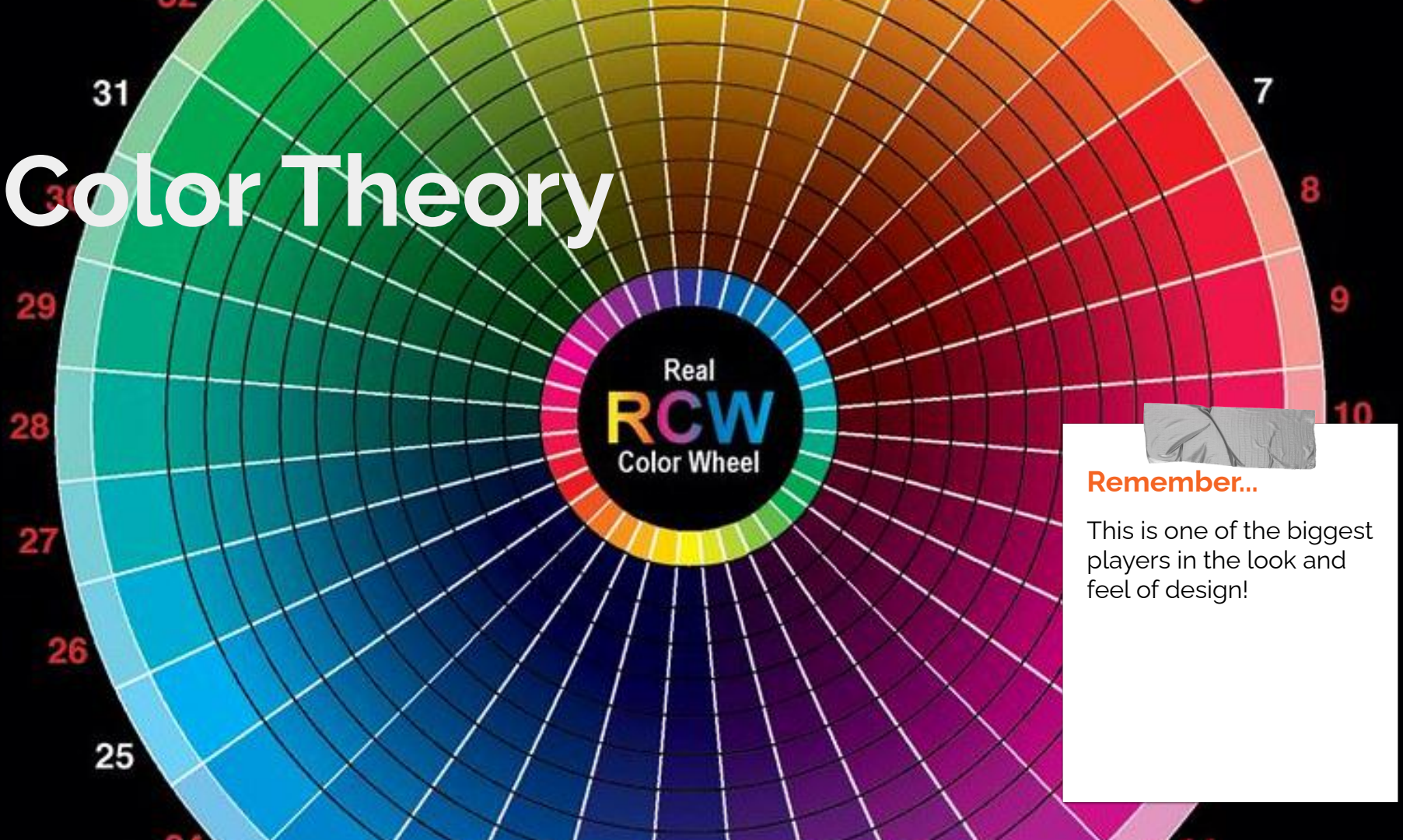
Let's Get Started on Foundation of Web Development 101



Tip

Bare with me! This information may SEEM elementary but it is an EXTREMELY IMPORTANT concept to web development and design.

Color Theory



Remember...

This is one of the biggest players in the look and feel of design!

The Color Wheel and Color Theory

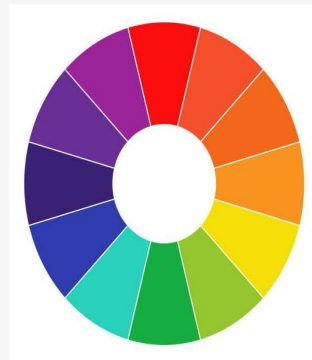
Color theory emerged when impressionist started experimenting with capturing light. In short, it boils down to how to human eye translates light with waves into color. Colors that match have similar complementary waves.

Primary Colors:

Red, Yellow, Blue

They are the basic colors of any color wheel and the color of spectrum as a whole.

To create and color, you'll have to use a combination of the primary colors.



The Color Wheel

Main Colors:

Red	Green
Orange	Blue
Yellow	Violet

In-Between Colors:

Red-Orange	Blue-Green
Yellow-Orange	Blue-Violet
Yellow-Green	Red-Violet

Secondary and Tertiary Colors

Secondary Colors:

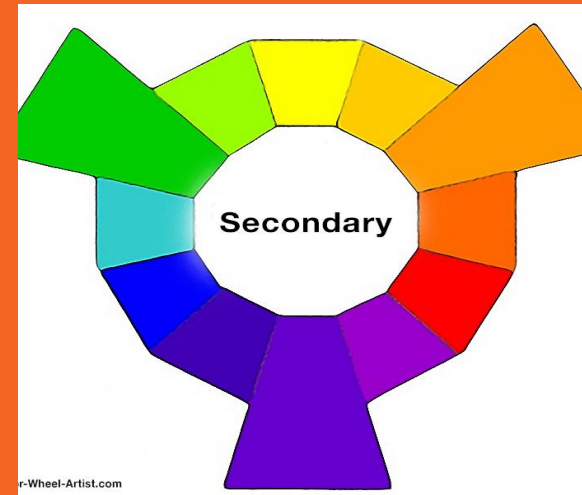
Orange, Green, Purple

1. Created with 2 primary colors
2. Lies opposite of primary colors
3. Opposite pairs are referred to as being complementary

Tertiary Colors:

1 primary + 1 secondary color

There are more tertiary colors than primary or secondary, owing to more combinations which can be made.



Warm & Cool Colors

Warm Colors:

Reds-Oranges-Yellows

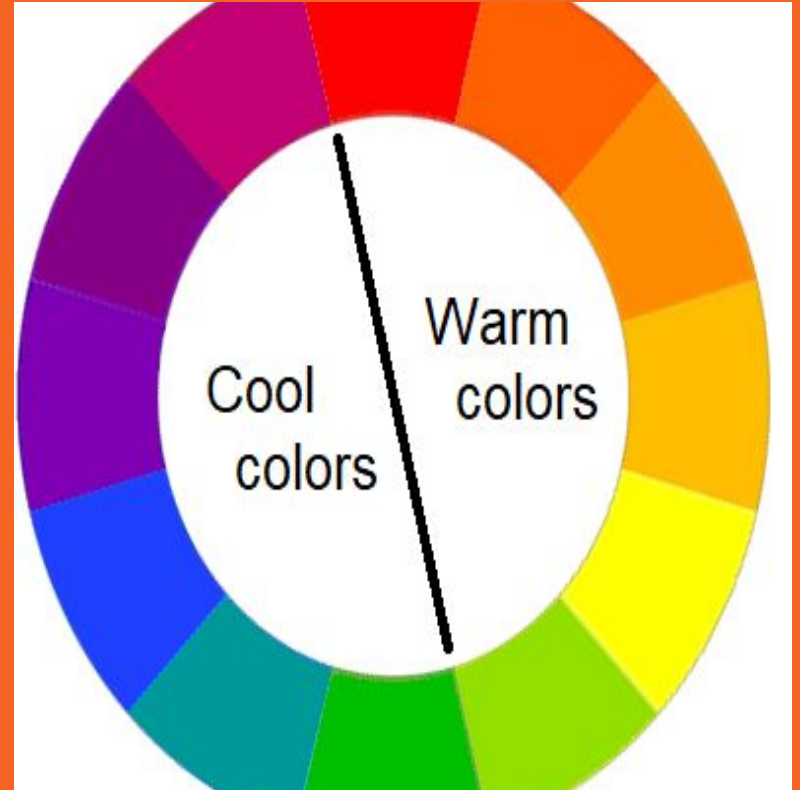
Evokes a feeling of warmth

Cool Colors:

Greens-Blues-Purples

Evokes a calm feeling. They are more subtle than warmer colors.

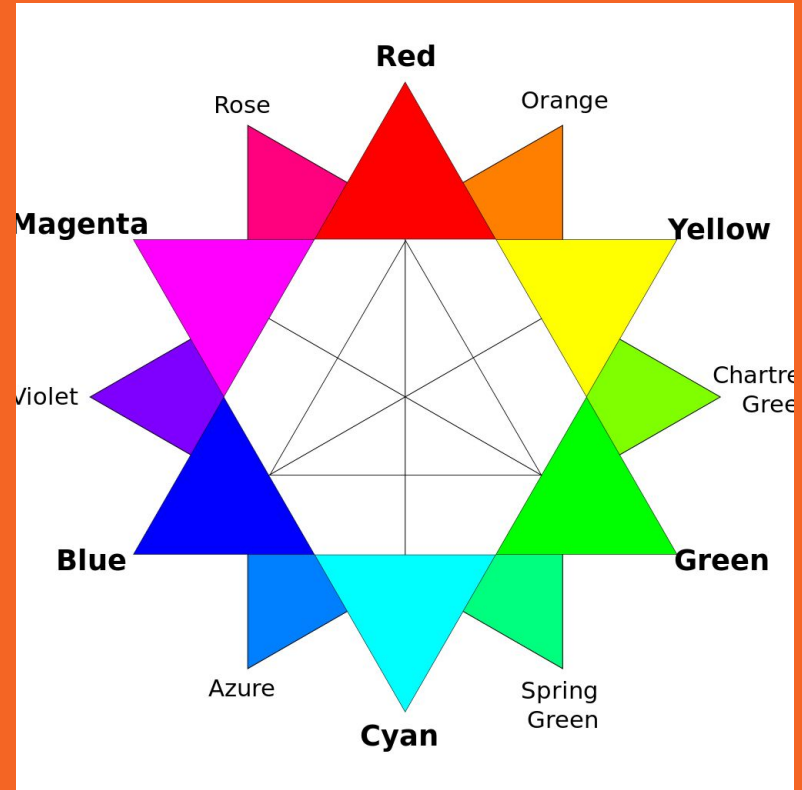
We'll get more into this in color schemes.



RGB: The Additive Color Model

- Red, green, and blue light are added together in various combinations to reproduce a wide spectrum of colors.
- LCD, plasma and LED displays all utilize RGB model.

We'll discuss this in more detail later in the course.





Now...

Color Schemes!

Color Schemes are an arrangement of colors that, once put together, can be used in any form of design.



Main Color Schemes

- Monochromatic
- Analogous
- Complementary
- Split Complementary
- Tradic
- Tetradic

We are going to cover the first three only... please visit [this](#) link to study the other color schemes.



Monochromatic

These schemes use one hue (such as red or blue) and then uses various tints or shades of that chosen hue.

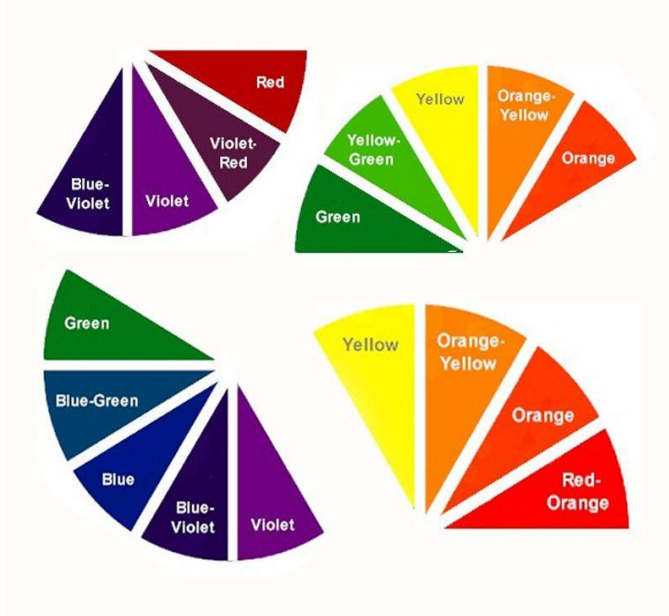
These are the easiest color schemes to create as it only involves that one hue - so you know that the few base colors you choose with work in harmony with one another.

Analogous

These schemes are created by choosing hues that are next to each other on the color wheel.

Example: green, yellow-green, yellow ect..

The easiest way to use an analogous color scheme is to choose one of the colors as the main, dominant color and then use the other two more sparsely, as accent colors to the first.



Complementary Color Scheme



Complementary

These schemes are created by choosing two colors opposite of each other on the color wheel.

Example: orange & purple, green & red

These colors offer high contrast and have a high impact. For this reason this color scheme can be somewhat hard to pull off. They work better when you have elements of a design you want to stand out, rather than using them in fuller doses across the whole design.

What's Next?... UX!

(User Experience)

User experience is a person's emotions and attitudes about using a particular product, system or service. Including practical, experiential, affective, meaningful and valuable aspects of human interaction.

[Understand more about UX](#)



Tip

Your design and the entire integration of your website needs to create a meaningful and relevant experience for the user.



UX for the web

- User Centered Design (UCD)
- Aesthetically may not always work with function/practically
- More complicated means less enjoyment
- The longer the user remains on a website/app the greater the chance of purchase decision or conversion. That is determined within the first 3 seconds.

Prime purpose is to provide a system to fulfill users needs. Mainly this is to interact with your product or service. (UI is not about eye candy!)

User Interface (UI)

UI : Practical to-dos

- Understand user behavior
- Influence the end-user
- Identify user needs (mobile?)
- Understand user wants



Best Practices

- Be consistent with design
- Easy to navigate layout/design
- Communicate clearly.(messages and labels)

Include key principles, questions, and elements to keep in mind.

User Centered Design (UCD)

UCD: Standards 6 Key Principles

- Design is based upon an explicit understanding of users, tasks, and environments.
- User test runs throughout design and development
- Design is driven and refined by user centered evaluations.
- The processes is iterative.
- Design addresses the whole user experience.
- Design includes multidisciplinary skills and perspectives.



UCD Questions

- Who is the targeted audience? (the users of the site)
- What is the main tasks and goals of the users
- What functions do the users need from the site?
- What information might the user need, and in what form do they need it?
- How does the targeted audience think the site should work?
- What are the extreme environments?
- Does the GUI utilize different input modes such as touching, speaking, gestures, or orientation?

What Is Website Usability?

The usability of a website tells us how effectively, and satisfactory its visitors or users can see, or examine by other means, the website.

This includes everything a user would typically experience when they visit the site including:

- Navigation bars, menus
- Content
- Images
- Videos
- Hyperlinks
- Buttons
- Forms
- Games



Usability Checklist

- **Page Load Time**
Optimizing your pages is IMPORTANT!
- **Clear Purpose**
The purpose of the page and critical actions are clear within 5 seconds
- **Logo/ Home links**
Logo is easy to find and links to the home page
- **Easy "About" Info**
Easy to find about, contact, and home information and links.
- **Consistency**
Be consistent throughout your design. Layout, navigation, search, logo etc.

That Was A Lot To Cover!!!

Let's move on and
talk about basic
photo editing!

[Photopea](#)