



# Interaction Design & HTML Basics

**DECO 1400/7140 Intro  
to Web Design**

University of Queensland

---

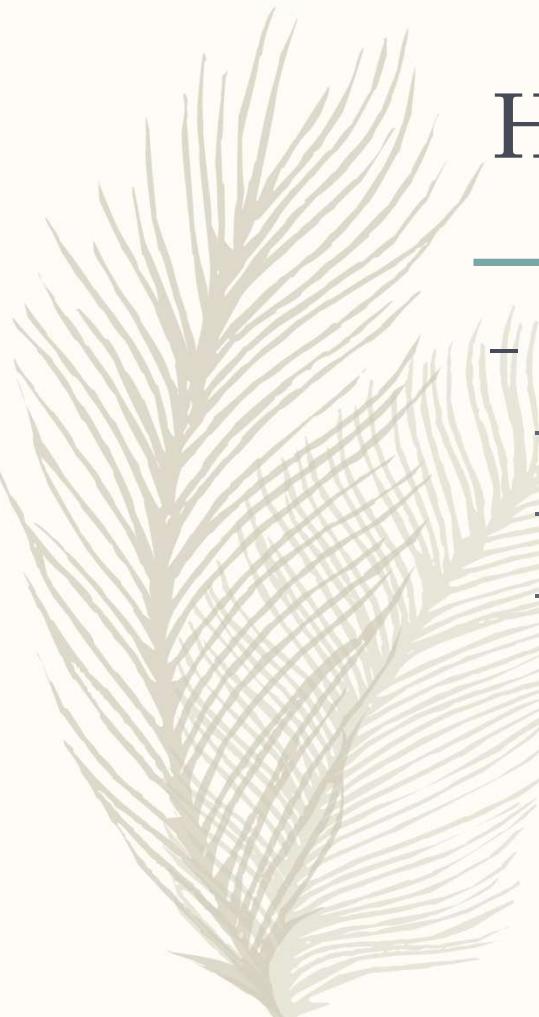
Semester 1, 2022



# Website Design and Implementation Assessments

---

- Out now... check Blackboard ☺
- Majority of the class are new to HTML5/CSS/JS based on the survey we did last week
- Releasing the main assignment early for those of you who are familiar with web design and/or programming
- Existing libraries: I would not say no, but we don't encourage you either!



# How to design?

---

- In deciding how to design your website, you should consider:
  - Who is going to be using the website
  - How they are going to it
  - And the sorts of activities will be carried out with the website



# Goals of interaction design

---

- Develop usable interactive products:
  - *easy to learn,*
  - *effective to use and*
  - *provides an enjoyable experience*
- Involve users in all stages of the design process (requirement gathering, design, pilot studies)

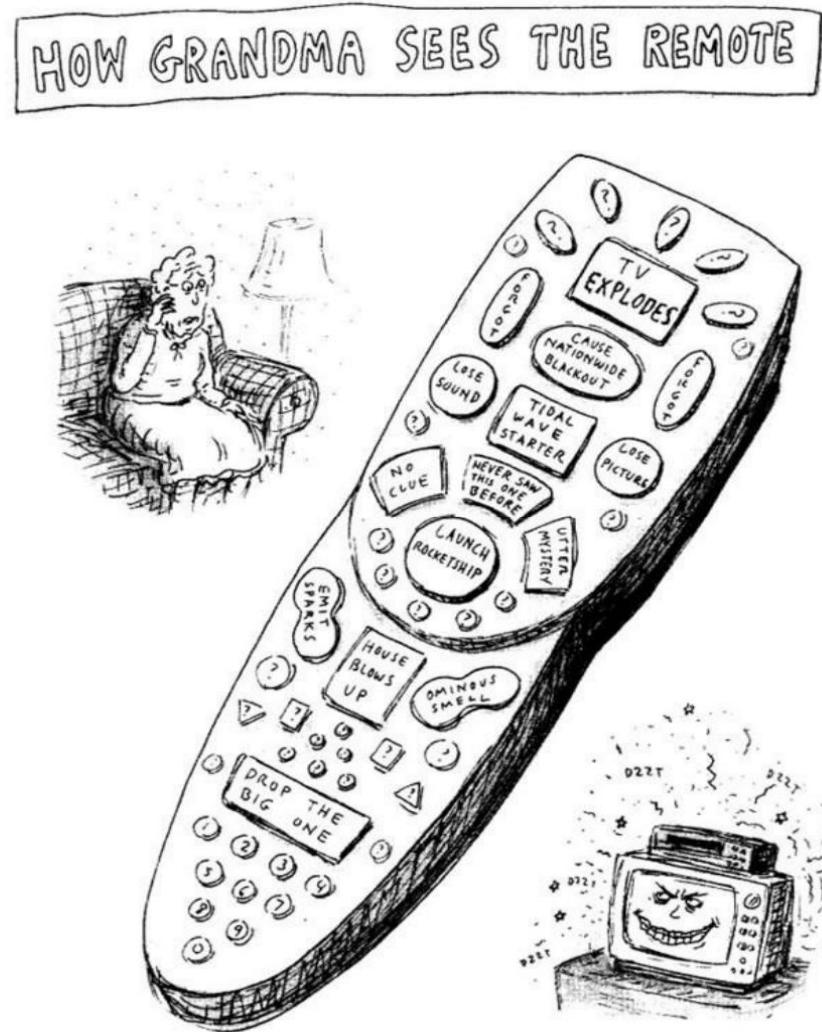
# Good and Bad design



- What is wrong with the Apex remote?
- Why is the TiVo remote so much better designed?
- What about AppleTV remote?

## UI Concerns

- Usability





<https://www.interaction-design.org>



© Boredpanda.com



It's not what it looks like...  
It's what it does.  
It's what it does to you.  
It's what it does to your job.  
It's what it does to your  
family.  
It's what it does to your  
friends.  
It's what it does to your  
hobbies.  
It's what it does to your  
leisure time.  
It's what it does to your  
health.  
It's what it does to your  
relationships.  
It's what it does to your  
sense of self-worth.  
It's what it does to your  
ability to succeed.  
It's what it does to your  
ability to be happy.  
It's what it does to your  
ability to live a fulfilling  
life.  
<https://www.interactiondesign.org/literature/article/the-invisible-hand>





# What I encountered in a recent overseas trip (pre-pandemic)...

---



# My recent experience

The screenshot shows the ROVER website homepage. At the top left is the Australian Government logo and the text "Australian Government Department of Infrastructure, Transport, Regional Development and Communications". Next to it is the ROVER logo with the text "Road Vehicle Regulator". To the right are two circular images: one showing a white van and another showing a hand holding a steering wheel. Further right is a house icon with the word "ROVER". On the far right are "Log Out" and "NB" buttons. Below the header is a navigation bar with links: Website, Legislation, Approvals, Payments, Authority to act, Notifications, Recalls, and Profile. Under "Approvals", there are icons for a globe, a gavel, a checkmark, a document with a dollar sign, a hierarchy, a bell, a warning sign, and a person. At the bottom left are buttons for "My Applications" and "New Applications".

Australian Government  
Department of Infrastructure, Transport,  
Regional Development and Communications

**ROVER**  
Road Vehicle Regulator

Website Legislation Approvals Payments Authority to act Notifications Recalls Profile

My Applications New Applications

Department of Infrastructure, Transport,  
Regional Development and Communications

**ROVER**  
Road Vehicle Regulator

Application - Single Road Vehicle

Log Out

Before you begin

Applicant details

Contributors

Vehicle use

**Vehicle details**

Required supporting material - Personal Effects

Contraventions

Comments and any other matters

Declarations

**Vehicle details**

What type of vehicle are you seeking approval for? \*

Passenger car or van (9 seats or less)

Please specify the make and model of the vehicle:

Make \*

Model \*

When was the vehicle manufactured?

Year: \* Month: \*

Please provide either the VIN or the chassis number for the vehicle. \*

VIN: Chassis number:

Vehicle seating capacity

Are you/the applicant the owner of the vehicle? \*

Yes  
 No, I am in the process of purchasing the vehicle  
 No, I am not the owner of the vehicle

**Edit**

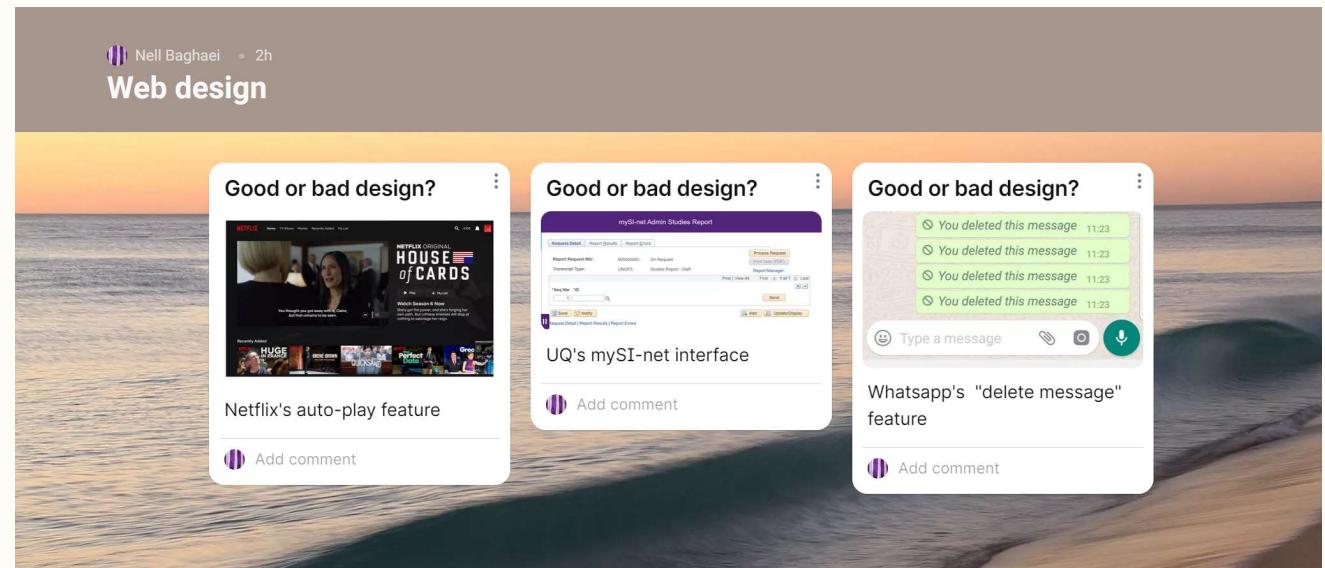
CRE-2021-0005973

<https://rover.infrastructure.gov.au/Produce//launch.aspx?id=95e3fb1c-4176-...>

# Padlet Activity

---

<https://padletuq.padlet.org/nbaghaei/goodorbaddesign>





# Ten Usability Heuristics for User Interface Design

---

- #1: Visibility of system status
- #2: Match between system and the real world
- #3: User control and freedom
- #4: Consistency and standards
- #5: Error prevention
- #6: Recognition rather than recall
- #7: Flexibility and efficiency of use
- #8: Aesthetic and minimalist design
- #9: Help users recognize, diagnose, and recover from errors
- #10: Help and documentation





in® SlideShare | Search

Home Explore Presentation Courses

Jakob Nielsen's

# 10 Usability Heuristics Explained

<https://www.slideshare.net/crafted/10-usability-heuristics-explained>

craft

August 2014.

1 of 32

Share icon



# 10 Usability Heuristics for User Interface Design

---

## #1: Visibility of system status

- The system should always keep users informed about what is going on, through appropriate feedback within reasonable time.

## #2: Match between system and the real world

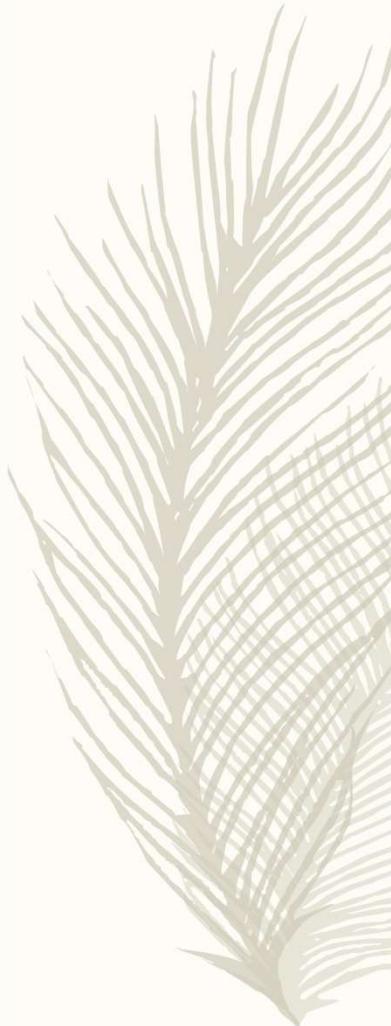
- The system should speak the users' language, with words, phrases and concepts familiar to the user, rather than system-oriented terms. Follow real-world conventions, making information appear in a natural and logical order.

## #3: User control and freedom

- Users often choose system functions by mistake and will need a clearly marked "emergency exit" to leave the unwanted state without having to go through an extended dialogue. Support undo and redo.

## #4: Consistency and standards

- Users should not have to wonder whether different words, situations, or actions mean the same thing.



## **#5: Error prevention**

- Even better than good error messages is a careful design which prevents a problem from occurring in the first place. Either eliminate error-prone conditions or check for them and present users with a confirmation option before they commit to the action.

## **#6: Recognition rather than recall**

---

- Minimize the user's memory load by making objects, actions, and options visible. The user should not have to remember information from one part of the dialogue to another. Instructions for use of the system should be visible or easily retrievable whenever appropriate.

## **#7: Flexibility and efficiency of use**

- Accelerators — unseen by the novice user — may often speed up the interaction for the expert user such that the system can cater to both inexperienced and experienced users. Allow users to tailor frequent actions.

## **#8: Aesthetic and minimalist design**

- Dialogues should not contain information which is irrelevant or rarely needed. Every extra unit of information in a dialogue competes with the relevant units of information and diminishes their relative visibility.

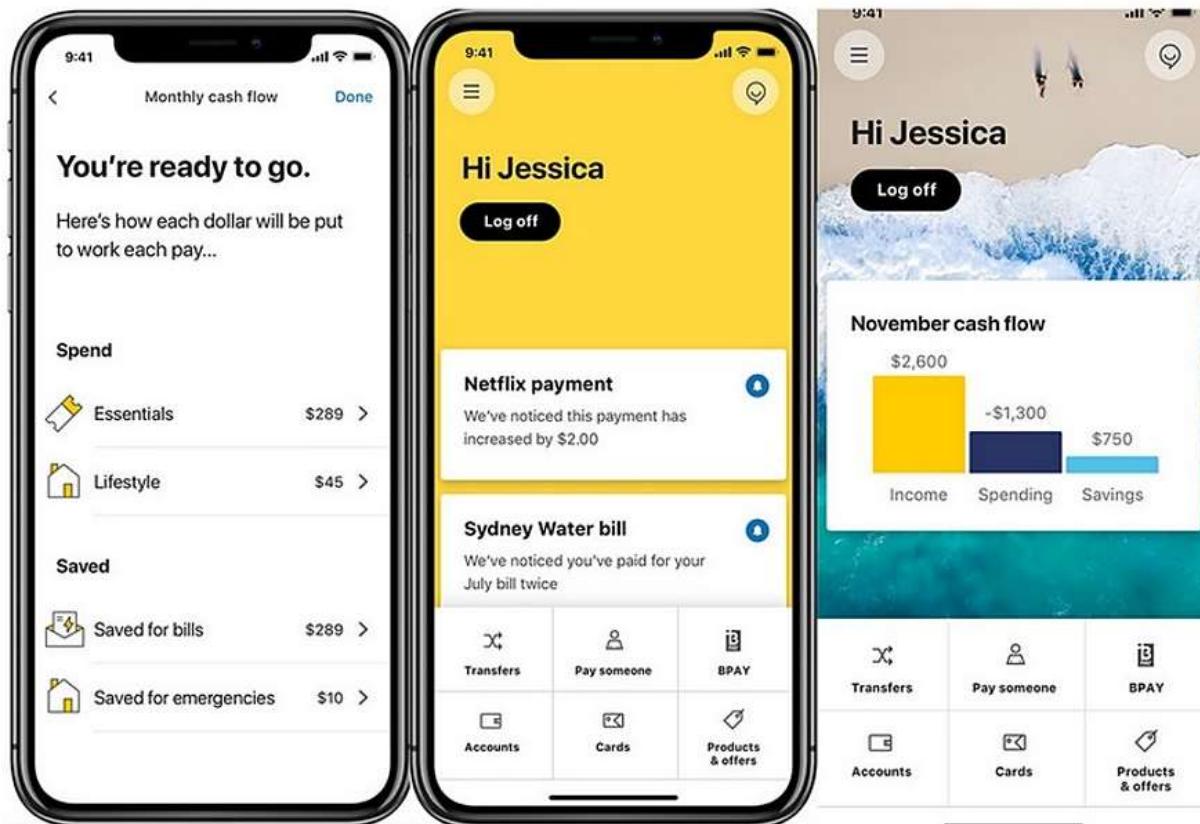
## **#9: Help users recognize, diagnose, and recover from errors**

- Error messages should be expressed in plain language (no codes), precisely indicate the problem, and constructively suggest a solution.

## **#10: Help and documentation**

- Even though it is better if the system can be used without documentation, it may be necessary to provide help and documentation. Any such information should be easy to search, focused on the user's task, list concrete steps to be carried out, and not be too large.

# Commonwealth Bank mobile app



Stay in control of your spend  
& card security

Next

# commended Reading

## Heuristic Evaluation of User Interfaces

<https://www.nngroup.com/articles/ten-usability-heuristics/>

**Summary:** Jakob Nielsen explains the heuristic evaluation method, which allows you to judge a user interface design based on 10 well-proven general principles for human-computer interaction.



# 10 Usability Heuristics applied to Web design

---

<http://designingwebinterfaces.com/6-tips-for-a-great-flex-ux-part-5>



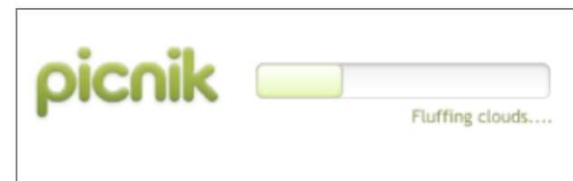
## 1. Visibility of system status (Feedback)

The system should always keep users informed about what is going on, through appropriate feedback within reasonable time.



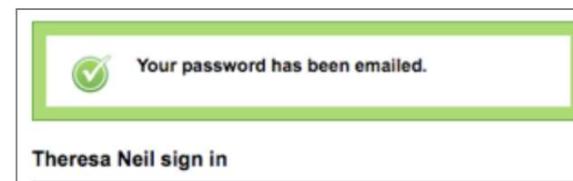
### 1.0 BaseCamp by 37signals

The upload button is enabled, until clicked. Then it is replaced with a progress indicator until the file has finished uploading



### 1.1 Picnik

Progress message and indicator shows while the application loads



We Bro  
Someth



A ninja stole this page.

You must return when the  
moon has friends and the  
fox is borrowed.

I

llection  
luding

# Managing User Experience - Medieval helpdesk



<https://www.youtube.com/watch?v=pQHX-SjgQvQ>



# How to learn more?

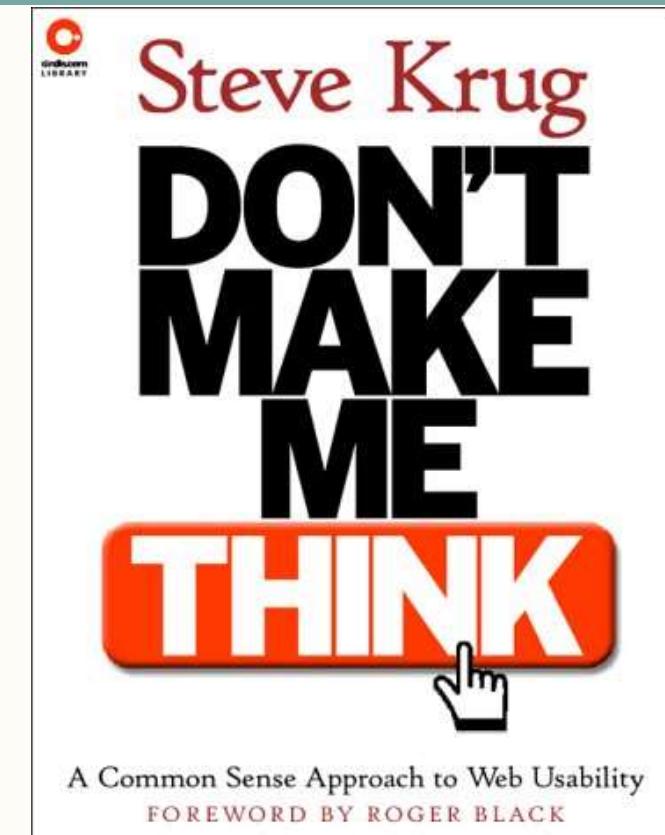
---

- The leading general journal for HCI is the [ACM Transactions on Computer-Human Interaction](#). However, there are many other well-established journals: [Human-Computer Interaction](#), [Interacting With Computers](#), [International Journal of Human-Computer Studies](#), [International Journal of Human-Computer Interaction](#), [Journal of Computer-Supported Cooperative Work](#), [Virtual Reality \(Springer\)](#)
- Top Conferences: CHI, OZCHI, ISMAR, INTERACT, etc.
- <https://www.interaction-design.org/literature/book/the-encyclopedia-of-human-computer-interaction-2nd-ed>

# Recommended Reading

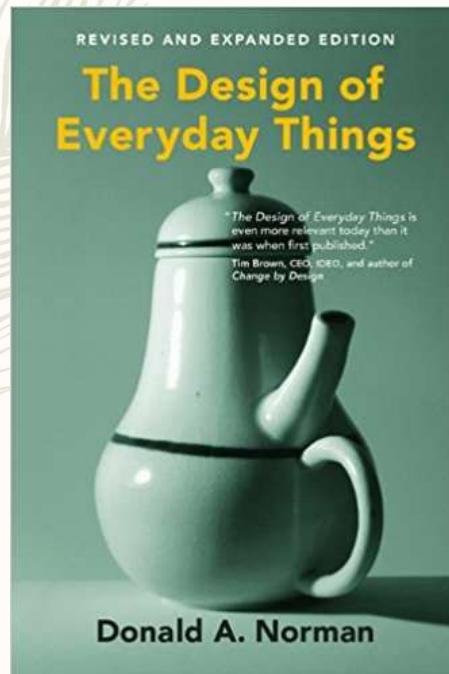
---

- Don't Make Me Think! A Common Sense Approach to Web Usability
  - explains the mental process going on in a user's head, and the best way to display information



# Recommended Reading

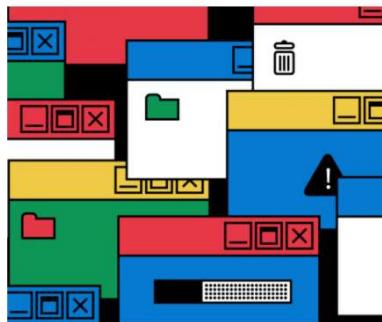
---



<https://www.nixdell.com/classes/HCI-and-Design-Spring-2017/The-Design-of-Everyday-Things-Revised-and-Expanded-Edition.pdf>

# Recommended Reading

---



## What is Graphical User Interface Design?

BY NICK BABICH

GUI is integral for creating effective human–



## Human Factor Principles in UX De

BY NICK BABICH

Familiar with human factors? Learn



## What Is Extended Reality (XR) Technology?



## Design Psychology: 6 Concepts Every UX Designer Should Know



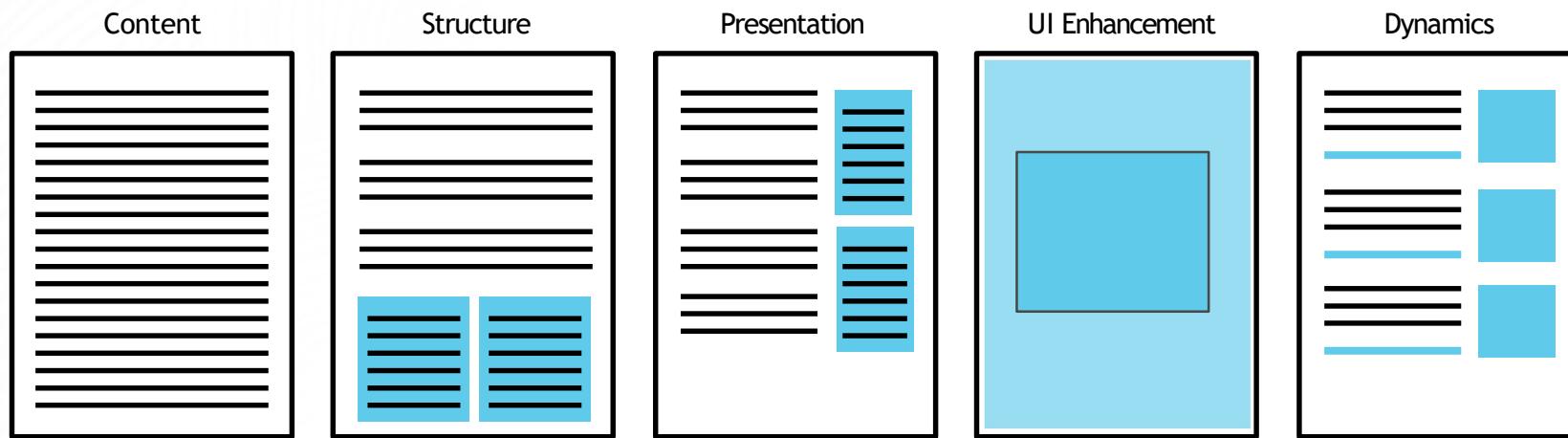
## Man and Machine: A Guide to Human–Computer Interaction

<https://xd.adobe.com/ideas/principles/human-computer-interaction/>

# HTML BASICS

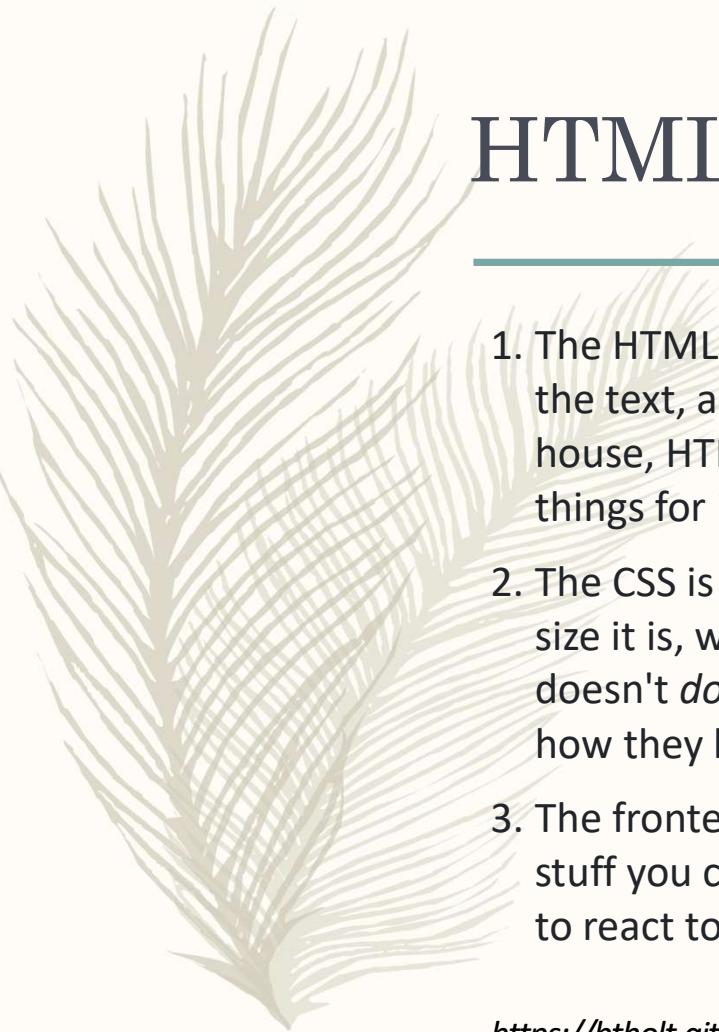
if (currentInteractionState > 0) {

# Creating a Website



<https://btholt.github.io/intro-to-web-dev-v2/overview>

Some of this content has been adapted from [Intro to Web Design](#), developed by Matt Luscombe



# HTML, CSS, JS

---

1. The HTML is the *structure*. It's going to contain all the text, the various images tied to the text, and it will generally group things together. Just like the drywall in your house, HTML doesn't do anything other than exist. It relies on other things to do things for it and to it.
2. The CSS is the *blueprint*. It's all the rules of what goes where, what color it is, what size it is, what font it is, what the decorative background images are, like HTML, CSS doesn't *do* anything, it's just a set of rules that describe what things go where and how they look.
3. The frontend / client JavaScript is the *smart home*. It's all the cool pre-programmed stuff you can tell your house to do. JavaScript is what can change the HTML and CSS to react to various stimuli.

<https://btholt.github.io/intro-to-web-dev-v2/overview>

Fixed content

Same for every visitor

<https://adventure.howstuffworks.com/10-best-outdoor-towns.htm>

Static Content

Changeable content

Different users, time of day, location, task

Requires special software to generate  
(usually on server)

<https://my.uq.edu.au/programs-courses/search.html?keywords=deco7140&searchType=all&archived=true>

# Dynamic Content

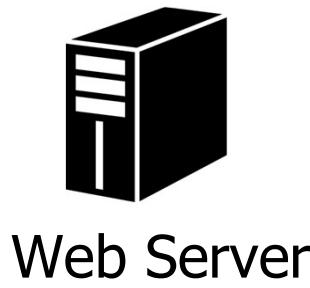
### Client-Side (user):

- Rendered by the browser
- All operations occur on users device
- HTML, CSS, JavaScript

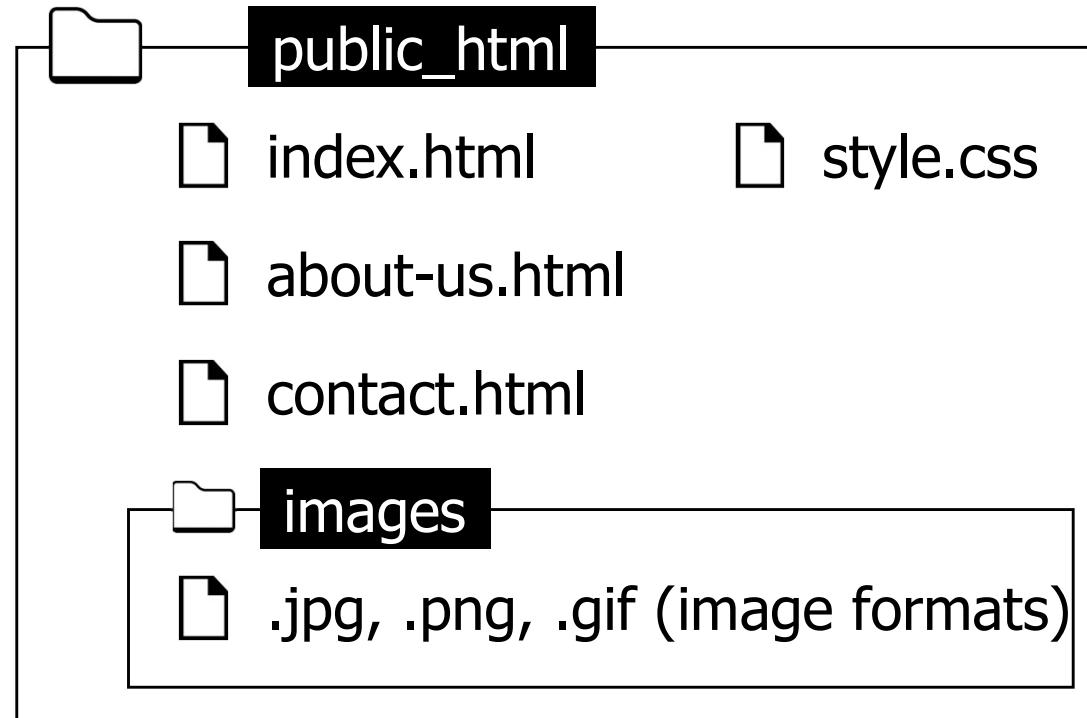
### Server-Side (hosting):

- Data retrieval, processing, storage performed by the server
- Generates unique code that is sent to the browser
- PHP, .NET, Java + more

# Client vs Server



Web Server



# Creating a Website

**HTML stands for:**

**Hyper Text Markup Language:**

- Identifies & structures content of a webpage
- HTML elements identify content
- Web browser interprets elements & renders on screen

**What is HTML?**

HTML documents == webpages

Plain text files saved with .html file extension

Files typically named use lowercase characters (a-z 0-9 - \_)

— Example: **description-of-page.html**

Edited using plain text, code editor:

— Notepad/TextEdit (not recommended),  
Visual Studio Code, Sublime Text etc

Viewed/rendered using web browser

# .HTML Files

Can I use [Dreamweaver, Squarespace, Wix etc] to build my website?

WYSIWYG ("What You See Is What You Get")  
software that generates HTML for you

They are for people who can't build websites

They usually add 'junk' to your HTML (we can tell if you've used one)

They can be difficult to debug

If YOU don't write the code,  
you CAN'T demonstrate learning objectives!

# Can I Use WYSIWYG?



Series of tags to describe content

A tag is a descriptor contained within angle brackets (or less than, greater than symbols):

```
<html>  
<p>  
<h1>
```

Tells the browser how to interpret the content following...

# HTML Syntax

Elements (and tags) are the building blocks of HTML

Elements consist of an *opening tag*, *some content* and a *closing tag*

A tag describes the content contained within

# HTML Elements

**Opening Tag**

**Content**

**Closing Tag**

`<h1> Welcome </h1>`

`<p> Some text here! </p>`

# HTML Elements

Some elements have no “content”

Or the “content” is described in attributes (for images/media)

Closing is denoted by no end tag

# HTML Elements

**Opening Tag**

**Content**

**Closing Tag**

```

```

```
<br>
```

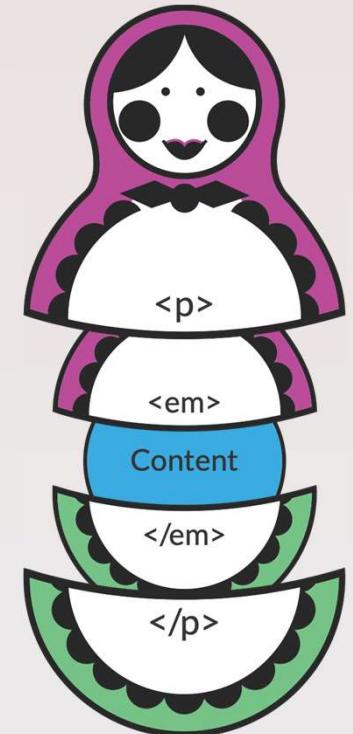
# HTML Elements

Elements can contain  
other elements

Tags must end in order

`<p>This is <em>Content</em></p>`

Nesting Elements



All HTML documents follow the same page structure:

```
<!doctype html>
<html>
    <head></head>
    <body></body>
</html>
```

# Basic Page Structure



```
<!doctype html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <title>Title of Page</title>
  </head>
  <body>
    Content of document...
  </body>
</html>
```

# Must Specify a Doctype

Doctype == Document Type Declaration

Always the first line in a HTML document

Instructs browser what rules to use in interpreting HTML

Specifies what version of HTML being used:

```
<!doctype html>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
```

# Doctype?

HTML5! Latest spec for HTML from the W3C:

```
<!doctype html>
```

HTML5 is a loose standard:

- Pro: it's backwards compatible and forgiving
- Con: it's easier to develop bad habits
- It tries to validate anything & everything

Which Version of HTML?

We will use HTML5 but with some rules:

- All elements must be closed
- All tags should be in lowercase
- All HTML documents must have an `html`, `head` & `body` element
- All text should be contained in the appropriate element
- Attribute values should always be placed within quotes

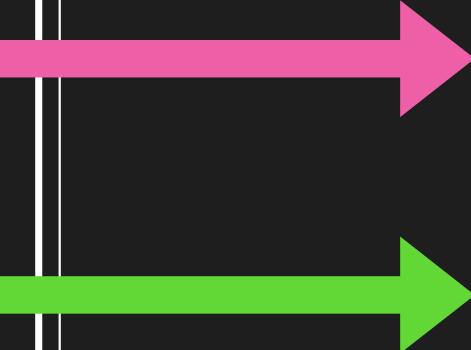
# HTML5 + Good Practice



```
<!doctype html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <title>Title of Page</title>
  </head>
  <body>
    Content of document...
  </body>
</html>
```

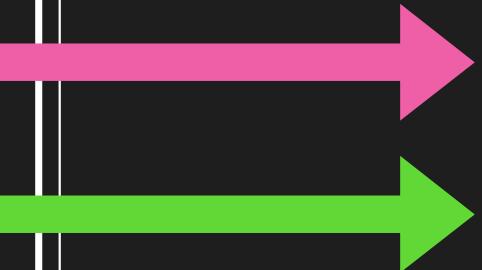


# Only 1 HTML Element



```
<!doctype html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <title>Title of Page</title>
  </head>
  <body>
    Content of document...
  </body>
</html>
```

# Only 1 Head Element



```
<!doctype html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <title>Title of Page</title>
  </head>
  <body>
    Content of document...
  </body>
</html>
```

# Only 1 Body Element

You can add comments to your HTML

These are not displayed on the page

FYI for yourself & other developers:

```
<!-- Comment Here -->
```

# HTML Comments

```
<!doctype html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <title>Title of Page</title>
  </head>
  <!-- Start of Content -->
  <body>
    Content of document...
  </body>
</html>
```

# Good practice HTML!

Each HTML element must abide by its content rules:

[https://developer.mozilla.org/en-US/docs/Web/Guide/HTML/Content\\_categories](https://developer.mozilla.org/en-US/docs/Web/Guide/HTML/Content_categories)

# Semantics

HTML can be submitted to a validator to check for errors  
W3C Markup Validation Service picks up structural or semantic  
issues with your code:  
<https://validator.w3.org>

Use your eyes too! The validator doesn't pick everything up

## Checking Code Quality

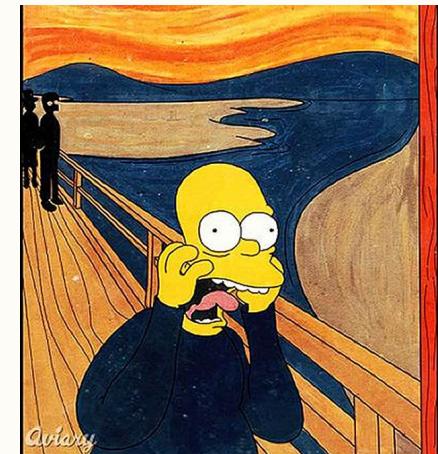
*Demo*



# DECO 7140: Introducing Assignment 1

---

- Available on Blackboard
- Worth 20% of your final grade
- Due Date: **Fri 8th April, 1:00pm**
- Upload your submissions on Blackboard
- Please talk to your tutor/myself, if you need help or post your question on Ed Discussion
- Don't leave it until the last minute!



# Tutorial

---

In this activity, you will form groups of 2-4 and look up a few examples of good and bad designs (refer to Goals of Interaction Design, Slide 3). The product can be a website, a mobile app or any other product. You will then present your findings (design examples and justifications) to the class.

Contribute to PeerWise by posting/responding/rating at least a question.





# What now?

---

- Make sure you attend your 3-hour slot of Tutorial/Practical every week (if you are an *external* student, that will be on Zoom)
- Check your Tutor's link on Blackboard, if you are in an online session. In-person classes are in 78-109.
- Start thinking about your assignment
- If you have any questions about the course or the materials, post them on *Ed Discussion Board*
- Check *Blackboard* for information and resources every week ☺