Web Development and the (HTML) Element of Competition

Katie Bell
https://groklearning.com/
A BIT ABOUT ME

- University
- NCSS/GPN
- Google
- Google (Switzerland)
- Grok Learning
TALK PLAN

1. A brief history of the internet
   (and being a web developer)
2. Where web development fits in the curriculum
3. Web.Comp
4. Showing off the students’ work
A BRIEF HISTORY OF THE INTERNET
A BRIEF HISTORY OF THE INTERNET WORLD WIDE WEB
EARLY DAYS (1990 - 1993)

Welcome to the GNU Project web server, www.gnu.org. The GNU Project was launched in 1984 to develop a complete UNIX style operating system which is free software. The GNU system, GNU is a recursive acronym for GNU's Not UNIX (GNU). It is pronounced "Gee-Oh-Neen" (GNU). Variants of the GNU operating system, which use the Korn/System V, are now available. Although those systems are often referred to as "GNU/Linux", they are more accurately called "GNU/Linux" systems.

This is also the web site of the Free Software Foundation (FSF). FSF is the principal organizational sponsor of the GNU Project. FSF receives very little funding from corporations or grant-making foundations. We rely on support from individuals like you who support FSF's mission to preserve, protect, and promote the freedom to study, copy, modify, and redistribute computer software, and to defend the rights of Free Software developers.

This is a new version of the NextStep WorldWideWeb application with the Link/4W library. Bug reports to mail@w3.org, quoting the version information above. Check the list of known bugs in the web text.

This was the original prototype for the World Wide Web. Many browsers for other platforms now exist. Read the web for details. After many years of testing, this application has now shipped images and marked HTML elements and flags. If you have an Internet connection, then you may try using "Help" under the Info menu to allow you all about this application. If you don't have an Internet connection -- get one!
PRESENTATIONAL MARKUP -> CSS (1995)

```html
<h1><font color="red"> Heading </font></h1>
```

Content

```html
<h1> Heading </h1>
```

Presentation

```css
h1 {
    color: red;
}
```
STANDARDISATION & THE W3C

1994 - HTML2
1995 - Draft HTML3
1996 - CSS1
1997 - HTML 4
1998 - CSS2
2011 - CSS2.1
2011-??? - CSS3
2014 - HTML5
STANDARDISATION & THE W3C

1994 - HTML2
1995 - Draft HTML3
1996 - CSS1
1997 - HTML 4
1998 - CSS2
2011 - CSS2.1
2011-?? - CSS3
2014 - HTML5
MODERN WEB DEVELOPMENT

- GUI editors? (E.g. Frontpage/Dreamweaver)
- Separation of Style and Content
- Responsive design
- Server generated HTML
- Client generated HTML
WHERE IT FITS

Web development in the national curriculum
# Representation of Data

<table>
<thead>
<tr>
<th>Stage</th>
<th>Learning Objective</th>
</tr>
</thead>
<tbody>
<tr>
<td>K-2</td>
<td>Recognise and explore patterns in data and represent data as pictures, symbols and diagrams (ACTDIK002)</td>
</tr>
<tr>
<td>3-4</td>
<td>Recognise different types of data and explore how the same data can be represented in different ways (ACTDIK008)</td>
</tr>
<tr>
<td>5-6</td>
<td>Examine how whole numbers are used to represent all data in digital systems (ACTDIK015)</td>
</tr>
<tr>
<td>7-8</td>
<td>Investigate how digital systems represent text, image and audio data in binary (ACTDIK024)</td>
</tr>
<tr>
<td>9-10</td>
<td>Analyse simple compression of data and how content data are separated from presentation (ACTDIK035)</td>
</tr>
</tbody>
</table>
In this next example, we can enter a negative number for all colour channels to reduce the brightness evenly across all colours.

File name: dragonfly.png
Red tint: -50
Green tint: 0
Blue tint: 0
REPRESENTATION OF DATA

K-2
Recognise and explore patterns in data and represent data as pictures, symbols and diagrams (ACTDIK002)

3-4
Recognise different types of data and explore how the same data can be represented in different ways (ACTDIK008)

5-6
Examine how whole numbers are used to represent all data in digital systems (ACTDIK015)

7-8
Investigate how digital systems represent text, image and audio data in binary (ACTDIK024)

9-10
Analyse simple compression of data and how content data are separated from presentation (ACTDIK035)
<table>
<thead>
<tr>
<th>5-6</th>
<th>Design a user interface for a digital system (ACTDIP018)</th>
<th>Design, modify and follow simple algorithms involving sequences of steps, branching, and iteration (repetition) (ACTDIP019)</th>
</tr>
</thead>
<tbody>
<tr>
<td>7-8</td>
<td>Design the user experience of a digital system, generating, evaluating and communicating alternative designs (ACTDIP028)</td>
<td>Design algorithms represented diagrammatically and in English, and trace algorithms to predict output for a given input and to identify errors (ACTDIP029)</td>
</tr>
<tr>
<td>9-10</td>
<td>Design the user experience of a digital system by evaluating alternative designs against criteria including functionality, accessibility, usability, and aesthetics (ACTDIP039)</td>
<td>Design algorithms represented diagrammatically and in structured English and validate algorithms and programs through tracing and test cases (ACTDIP040)</td>
</tr>
</tbody>
</table>
Hello with attitude

Write a program that asks the user for their name, and then replies (with a bit of attitude). My interaction with your program should look like this:

```
What is your name? Grok
So you call yourself 'Grok' huh?
```

Watch out for the punctuation!

---

```
# Enter your code for "Hello with attitude" here.
name = input("What is your name? ")
print("So you call yourself " + name + " huu?"")
```

---

Submissions

- [x] #1 All tests passed!

- Testing the example in the question.
- Checking for the correct capitalisation in the example in the question.
- Checking for the correct punctuation in the example in the question.
- Checking for the correct whitespace in the example in the question.
- Testing with the input "Jen".
- Testing a two word name.
GENERATING AND DESIGNING

5-6
Design a user interface for a digital system (ACTDIP018)

7-8
Design the user experience of a digital system, generating, evaluating and communicating alternative designs (ACTDIP028)

9-10
Design the user experience of a digital system by evaluating alternative designs against criteria including functionality, accessibility, usability, and aesthetics (ACTDIP039)

Design, modify and follow simple algorithms involving sequences of steps, branching, and iteration (repetition) (ACTDIP019)

Design algorithms represented diagrammatically and in English, and trace algorithms to predict output for a given input and to identify errors (ACTDIP029)

Design algorithms represented diagrammatically and in structured English and validate algorithms and programs through tracing and test cases (ACTDIP040)
**Celestial Calendar**

Here's a website showing some of the celestial events for 2016, but it would be much better with a hero image and some styles.

**Add the image**

Add the image `starry-sky.jpg` before the text. It should be 75% of the width of the page, with the alt text `Amazing starry night sky`.

**Add your styles to `main.css`**

Make the body background colour black, and the body text color `rgb(238, 138, 90)` and centre aligned.

The result should look like this:

![Image of a starry sky]
HISTORY

- Web.Comp 2015 (June)
- Introduction to HTML/CSS
- Web.Comp 2016 (running now!)
GROK COURSE PRINCIPLES

- All-inclusive (no assumed knowledge)
- Learn by doing
- Instant feedback
GROK COURSE PRINCIPLES

- All-inclusive (no assumed knowledge)
- Learn by doing
- Instant feedback
- Instant helpful and actionable feedback
WEB.COMP PRINCIPLES

- Modern websites
- Official standards
- Real user interface design
  - Accessibility
  - Usability
  - Aesthetics
WHAT WE TEACH

○ HTML for content example
○ CSS for presentation example
○ How familiar websites are done example example
○ Complete sites example example
ACCESSIBILITY

- Semantic HTML for screen-readers
- Alt-text on all images
- Accessible colour schemes
Semantic HTML

The purpose of HTML is to represent the structure of a document, that is, to describe the meaning of each piece of text. So far you have learned to represent headings, paragraphs and lists. Now we'll look at the structure of the whole document.

Using a clear and logical structure in the whole document is important for both humans and computers to be able to read it.

It's especially important for vision-impaired users who use screen-reader software to browse the internet. Clear document structure helps them find the important content faster instead of listening through the heading and navigation links for every page they visit.

It also allows automated tools, such as search engines, to...
ACCESSIBILITY

- Semantic HTML for screen-readers
- Alt-text on all images
- Accessible colour schemes
<article>
  <p><img src="comic.jpg" alt="Man sees another person wearing the same shirt as him. Feeling uncomfortable, he quickly puts on a jacket to cover it."></p>
</article>
USABILITY

- Highlighting clickable elements
- Layout and hierarchy of information
- White-space and clutter
- Page loading times
AESTHETICS

- Fonts and colour schemes [example]
- Space and layout [example]
- Contrast [example]
THE TOURNAMENT

Now it’s your [students’] turn.
A DESIGN CHALLENGE

- One week
- One website (HTML provided)
- Full separation of content and presentation
THE RESULTS

ANY QUESTIONS?

Find me at:

○ https://groklearning.com/
○ Email: katie@groklearning.com
○ Twitter: @groklearning, @notsolonecoder
WHO USES GROK?

- **Schools around the world**
  - Some you’ve heard of: Christ Church Grammar, Knox, MLC, PLC, Queenwood, Abbotsleigh, Barker, Kambala, Sydney Boys, Gungahlin College, Tara, St Pius X, James Ruse, and many more.

- **A number of universities**
  - University of Melbourne, University of Sydney, Australian Catholic University

- **Thousands of Individual learners**
FOR TEACHERS

○ FREE access to all courses for teachers
  • (less than the cost of a textbook for students)
○ Professional Development workshops focused on Digital Technologies
Learn to code from your browser!

Start our Beginners Course

We cover an introduction to programming using Python 3. The first two modules of all our courses are available for free!

Learning to code has never been so easy or fun!

Challenging problems
Our problem-based learning approach matches each new concept with a problem so you learn by doing!

Code in the browser
Code from any device, any time - start a question on your iPad at school and finish it off on your computer at home.
Many-pointed snowflake

To show you how useful repeat blocks are, let’s draw a snowflake with 12 points like this!

This would take a really long time if we wrote out each instruction one by one, but with a repeating loop it’s a very short program and much easier to build!
What's the Time?

Write a program that asks the user for the current time in hours and minutes. Your program should then draw a clock face, the minute hand, and the hour hand. For example:

Hour: 4
Minute: 12

```python
from turtle import *

hours = int(input('Hour: '))
mins = int(input('Minute: '))

for i in range(12):
    penup()
    forward(100)
    pendown()
    forward(20)
    penup()
    backward(120)
    right(30)
penend()```

Run Terminal Save Copy

Mark What's the Time?
Hello with attitude

Write a program that asks the user for their name, and then replies (with a bit of attitude). My interaction with your program should look like this:

```
What is your name? Grok
So you call yourself 'Grok' huh?
```

Watch out for the punctuation!

Edit slide | Edit problem

Submissions

All tests passed!

- Testing the example in the question.
- Checking for the correct capitalisation in the example in the question.
- Checking for the correct punctuation in the example in the question.
- Checking for the correct whitespace in the example in the question.
- Testing with the input "Jen".
- Testing a two word name.