HTML & CSS

Together at last

Python Exercise – Angry Dice

- Let’s all play Angry Dice!

Agenda for Today

- Python Objects Recap
- Code Reviews
- Final Projects!
- Other CSS Selectors
- List styles
- Block vs Inline vs Inline Block
- Divs & Spans
- Navigation Exercise

Objects

- *def* defines a function, yes? So *class* defines an Object Class – a grouping of variables and functions
  - functions within a Class are sometimes called methods
- Classes aren't just a jumble of code – they're a logical grouping of items

Objects

A logical grouping?

- There is a logical connection between a class' variables and methods
- Die vs Card
- Person vs Account
- Blackjack vs AngryDice

Objects

- functions within a class should make sense in relationship to the object the class is defining
  - Die is to roll() as Card is to flip_over()
  - DiceBag is to shake() as Deck is to shuffle()

Functions & Variables

- think about how an Object breaks down into smaller parts, either through *function methods* or *instance variables*
  - Functions should only do one thing, and do it well
  - Instance variables should have information that the whole object needs to know about

Objects and Subclasses

Die vs AngryDie

Library Example

Using pen & paper and UML notation, design the objects needed to write a program that runs a Simple™ Library catalog (an old school one, with books). It only has books, and keeps track of these books. It needs to know:

- everything there is to know about the books
- where the books are shelved
- if a book is checked out, by whom was it checked out, and when it should be back by
- when a book is overdue
- when the last time a specific person checked out a book

There is no perfect answer

Give 200 programmers a programming assignment and you will get 83 different answers – roughly, depending on how they phrased their Google queries.

Code is cheap. Practice is priceless.

Final Projects!

Due next Friday!

Needs to:

- Demonstrate a complete understanding of Python, HTML, CSS, Java, C++, QBasic, Perl, ASP.Net, Binary, Klingon, x86 Assembly
- Follow PEP8, Closure, and SepTAC standards
- 10,000 line minimum, single spaced
- Designed so well that you're featured on the AppStore

Final Projects

I’m kidding.

Start thinking about what you want to make!

DO NOT think about HOW you want to make it – think about WHAT you want to make it.

CSS Selectors

You can select elements to style a variety of ways. The most basic ways are by:

- element
- class
- id
- combinations

Element

```
element_name {
    // Style for the element!
}
```

This method will style ALL of the element_name elements with the defined styles.

Class

```
<p class=”class_name”>Text!</p>


<style>
    .class_name {
        // All the styles for this class
    }
</style>
```

Id

```
<p id=”best_paragraph”>Text</p>
#id_name {
    // styles for the id’d element
}
```

- Id’s are ways of identifying specific components in HTML. They must be unique within a single web page.
- You can style by them, but this is generally a bad idea! You never want to get too specific in CSS
- Id’s are more useful for Javascript

Some combination

- You can combine style selectors to access things more specifically
- The more general you can style an element, the better.

```
.body_text p {
    // Style all the p tags within the body_text class
}
```

List Styles

- Lists are very powerful HTML elements, as they handle grouping and ordering things for you
- Styled right, lists can do a LOT of stuff
  - navigation
  - lists
  - image galleries
  - recipes
  - social media cards

List Styling

- list-style-type: circle/square/none
  - defined the bullets or numbers beside the list
- list-style-image:
  - uses an image as the bullet

Navigation Exercise

Add navigation to your portfolio page, linking to your other two webpages.

It should be a clearly defined nav block that demonstrates your understanding of list-styles, selectors, and block vs inline concepts.