Building a Web Page with HTML

* Tool used to make the web page: codepen.io
* Hyper Text Markup Language
  + HTML is a language used to make web pages
    - Not a programming language, markup language
    - Specifies contents of web pages, used by browsers
    - Specifies meaning or semantics
      * CSS helps how to display them
  + The web page is defined between the starting HTML tag and the ending HTML tag
    - End tag has a slash that indicates it’s the end tag
* Metadata and Sectioning Elements
  + Metadata elements
    - Information about the page
    - <html>
      * Contains all other elements
      * Specifies using HTML standard
    - <head>
      * Information about the page: title, scripts, CSS
    - <title>
      * Specifies page title
      * Nested inside <head> </head> tags
  + Sectioning elements
    - Define regions
    - <body>
      * Contains all items seen on page
    - <h1>
      * Section header
      * Also <h2>, <h3>… <h6>
    - <div>
      * Define section of web page
      * Useful for CSS
* Formatting Text and Nesting Tags
  + Tags surround text or page elements
* Adding Images and Links
  + Web page can include images, video and audio
    - Different kinds of tags
  + Image tag

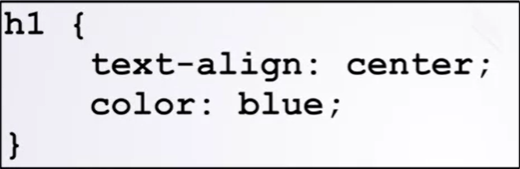
<img src = “…” width = “…”/>

* + - Options: extra information
    - No end tag, src required, with optional
      * Height=
      * CSS later
  + Linking pages together

<a href = “…”> … </a>

* + - Anchor tag
    - href attribute
    - clickable text
* Listing and tables
  + Simple Lists
    - Some lists use circles or bullets
      * Unordered lists, tag: <ul></ul>
      * Content viewed in order, list labels all the same
      * Inside <ul></ul> must have sequence of <li></li> elements.
        + Anything between <li></li>elements, not just text: <img>, <a href>, <ul>, …
      * Ordered list <ol>
        + <li></li> required
        + Automatic numbering
      * We can compose HTML elements
      * Nested lists
        + <ul> in <li></li>
  + HTML tables
    - Tables information in in rows and columns
    - HTML table elements
      * <table></table>
      * Contains rows <tr></tr>
    - Rows contain
      * Header elements <th></th>
      * Table cells <td></td>

Styling a Webpage with CSS

* Cascading Style Sheets
  + Specify look and formatting
  + Separate contents from presentation
    - How big is <h1>?
    - Color for <h1>?
    - Mobile or desktop?
  + Design that scales
    - 1000 pages on website
* CSS basics
  + CSS syntax
    - 
  + Class: named styles
    - HTML: <li class = a> apple</li>
    - CSS: .a{…}
  + ID: name one element
    - HTML: <img src = “…” id =a”>
    - CSS: #a{…}
  + Combinators: selected by relationship
    - Style <li> inside of <ul>:
      * ul li {…}
    - more advanced relationship exist
* Color and Names in CSS

Algorithms and Programming Concepts

* Computational thinking
  + Everything is a number
  + Abstraction
    - separation of interface + implementation
  + Green Screen
* A Seven Step Approach to Solving Programming Problems
  + Seven steps

1. Work example by hands
2. Write down what you did
3. Find patterns
4. Check by hand
5. Translate to code
6. Run test cases
7. Debug failed cases

Programming with Java Script

* Variables
  + Naming a variable
    - var x = 3;
      * Keyword “var”: declaring a variable
      * The name of the variable
      * Equal sign
      * The value to initialize the variable to
      * A semicolon: end the statement
  + Semantics: Meaning
  + Variable holding images
    - var fgImage = new SimpleImage(“drewrobert.png”);
      * Keyword “new”: “make an object”
      * Name of type to create
      * Parameters: specify more information
  + Methods
    - Perform some (complex) function
    - Act on an object
    - Can have parameters
    - Example: fgImage.getWidth();
  + Functions
    - Example: function square (x) {

var ans = x \* x;

return ans;

}

* For Loops
  + Repetition: for loops
    - Assuming img previously declared/initialized
  + Break down syntax
    - for (var pixel of img.values()){

var newG = 255 - pixel .getGreen();

pixel.setGreen(newG);

}

* + - Keyword for
    - Variable to refer to current item (looks like variable declaration)
    - Keyword of
    - What to repeat over

For images: imgvalues() give all pixels

* + - Body of the loop: what statement to repeat in curly braces
* Conditional Execution

JavaScript for Web Pages

* Event-Driven Programming
  + Buttons and Dives
    - Creating an HTML Button

<input type = “button” value = “change”

onclick = “alert (‘click button’)” >

* + - <input> tag
    - Type attribute
    - Type of input
    - Value attribute
    - Text on button
    - Onclick attribute: triggers event handler on click
    - Action for the button to do
  + Changing Pages Interactively
  + Getting HTML elements
    - Function changecolor(){

var dd1 = document.getElementById(“d1”);

var dd2 = document.getElementById(“d2”);

dd1.className = “blueback”;

dd2.className = “yellowback”;

}

* + - getElementBYId: access an HTML element using its associated ID, document refers the entire HTML web page
    - dd1.className: change the color of the div od the div with the ID d1
* Using HTML5 Canvas
  + <canvas> element
    - Used as part of DLTP and JavaScript image processing
    - Used in many other applications to display graphical context
  + Basic <canvas> functionality
* Input and Events
  + The HTML <input> element gets input from the user and process the input
    - Button
    - Text
    - Color picker
    - Range
  + Events
    - Mouse click
    - Mouse enter/leave
    - Field changes
    - Input given
  + Color picker
    - < input type=“color” value=“…”

id=“clr” onchange=“docolor()”>

* + - The color chooser’s value is accessed using the value. Attribute or field of the color chooser, HTML element stored in the variable color input
  + Slider input
    - <input type=“range” min=“10” max=“100”

value=“10” id=“sldr” oninput=“dosquare()”>

* + - ctx.clearRect(0,0,dd1.width,dd1.height);

clear the whole canvas

Green Screen Web Page

* Upload and Display an Image
  + Prototype to Simplify
    - <input type=“file” multiple=“false”

accept=“image/\*” id=“finput”

onchange=“upload()”>

* Convert Image to Grayscale
  + Global Variables
    - Defined outside all functions

Using BlueJ to Program in Java

* Java is object-oriented
* Code is organized in classes

Variables and Mathematical Operators

* Shapes: Collections of Points

Functions and Conditions

* Functions
  + Java does not have functions, it has methods, since all code in java is inside of objects.
* Conditionals
  + System.Out.printIn(): print something in java

Classes, Types, and For Each Loops

* Classes
  + Object oriented programming: group data and code into logical units called objects
  + Class: template that specifies how to make objects
    - public class Point

declaration for a class called Point

* + - private int x;

private int y;

declaration of two fields x and y, field is the name for a variable that is inside of an object. “private” means that only code inside the class can directly manipulate these fields.

* + - Public Point (int starx, int starty) {

x = startx;

y = starty;

}

Declaration of a constructor

* + - Methods are the functions inside of classes
    - Execution begins in main
  + Types
    - Types specify
      * Interpretation: what does a specific number mean?
      * Operation: what can you do?
    - Primitives vs Objects
      * Primitives: int, double, char, boolean, float, long, byte, short
        + Value is directly in the box
        + Can’t invoke methods on them
        + Can’t be null
      * Objects: String, Points, Shape, any class you write yourself
        + Reference(arrow) to object
        + Can invoke methods, access fields with dot
        + Can be null
        + == checks if arrows points at some objects
  + For each loops

Finding a Gene in DNA

* Understanding Strings
  + Numbers are stored in computer
    - In memory, on a flash-drive, on a hard drive
  + Information is often readable
    - We use strings to store readable data

Finding All Genes in DNA

* Conceptual Understanding
  + Searching for many genes
* While Loop Syntax and Semantics
  + while (x < y){

System.out.println(x);

x = x + 3

}

* + Generally, for “As long as (condition)”
* Three Stop Codons
  + TAA, TGA, TAG
* Logical And/ Or
  + And can be expressed by &&
  + Or can be expressed by ||
* Finding Multiple Genes

Using the StorageResource Class

* Separation of Concerns
  + Iterate over gene

1. Set startIndex to 0
2. Repeat following steps
3. Find the next gene after startIndex
4. If no gene was found, leave this loop
5. Do something with that gene
6. Set startIndex to just past the end of the gene
   * Could the algorithm be copy and be used with little changes

* StorageResource Class
  + What is StorageResource
    - Holds a collection of strings
      * Can .add(somestring) to put into it
      * Can iterate over it with .data()
      * Some other methods

StorageResource sr = new StorageResource ();

sr.add(“hello”);

sr.add(“world”);

for (string s: sr.data()){

system.out.println(s);

}

* + Gene Finding: Put in StorageResource

1. Create a empty StorageResource call it genelist
2. Set startIndex to 0
3. Repeat the following steps
4. Find the next gene after startIndex
5. If no gene was found, leave this loop
6. Add that gene to genelist
7. Set startIndex to just past the end of the gene
8. Your answer is the gene list

Which Countries Export...?

* CSV Data: Comma Separated Values
  + Tabular data: Common and Useful
    - Tabular(row/column) format data: write code
      * Csv files: comma separate values
  + Programming with csv files
    - Spread sheets extremely useful: visualizing, finding information, trends, much more.
    - Data in CSV format can be portable between different spreadsheets programs.
    - Common formats have standards
    - Use existing library: Apache CSVParser
* Using CSV Libraries
  + Favorite food and more
    - Different view of the food.csv data file
* Which Countries Export...? Developing an Algorithm
  + CSV data about exports
    - Country, exports, value
  + Problem: Which country exports some particular thing?
    - 4 countries, which exports coffee?
    - Look at the first row, “Exports” column
      * Did not contain “coffee”
    - Look at the second row, “Exports” column
      * Contains “coffee”
      * Wrote down “Madagascar”
    - Look at the third row, “Exports” column
      * Contains “coffee”
      * Wrote down “Malawi”
    - Look at the forth row, “Exports” column
      * Did not contain “coffee”
    - Find patterns, Generalize
      * For each row in CSV file
        + Look at the “Exports” column of that row
        + Check if it contains exportOfInterest

If so, write down the “country” from that row

Weather CSV Problem

* Hottest Day in a Year: Comma Separated Values
  + CSV data about weather at RDU
    - One file per day, one row per hour
    - Analyze and answer questions
    - Question: mas temperature?
      * One day: just look? Use MAX() in spreadsheet?
      * Many days?... need a program
    - Problem: find the hottest day in a day
      * Day with highest max temperature
      * Related: day with highest average temperature
    - Plan to write this program:
      * Learn about working with numeric data
      * Start smaller: max temperature on one day
        + Test, be confidence in code
      * Build on tested code: max temperature in year
* Converting Strings to Numbers
  + Numeric data in CSV file
  + Read in as String
    - Sequence of characters ‘1’ ‘4’ ‘9’ ‘3’
  + Want to work as int or double
  + String might not represent a number
  + Need explicitly convert
    - Algorithm: read character, produce int
  + Built into Java
    - Integer.parseInt
    - Double.pareDouble
      * Works for numbers with fractional part
* Maximum Temperature: Developing an Algorithm
  + Start with largestSoFar as nothing
  + For each row(currentRow) in the CSV file
    - If largestSofar is nothing
      * Update largestSoFar to be currentRow
    - Otherwise
      * Check if currentRow’s temperature > largestSoFar’s
        + If so update largestSoFar co currentRow
  + The largestSoFar is the answer

* + Java has null
    - Means “nothing” or “no such thing”
    - CSVRecord largestSoFar = null;
    - Return null;
    - If (largestSoFar == null)
  + No object: cannot call method on it
  + What type is null
    - Special null type
      * Type cannot be named in a program
      * Literal null has this type
      * Can be converted to any object type
    - Primitive Types: cannot be null
      * Int, double, char, Boolean
    - Object Types: can be null
      * FileResource, String, CSVRecord, Pixel…
      * Anything with methods
      * Any class you write