Front End Software Development

Introduction to JavaScript (weeks 1 - 6)

Week 06

Agenda

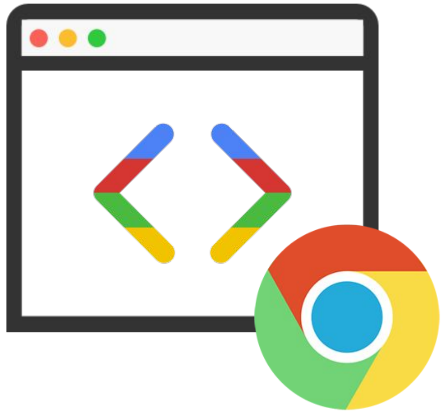
- Questions
- Chrome DevTools
- Debugging
- Unit Testing
  - Mocha and Chai
  - Test Driven Development
- Final Project

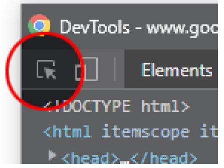
Questions

THERE'S A BUG IN MY COMPUTER?
BETTER SPRAY SOME RAID ON IT
imgflip.com

Chrome DevTools

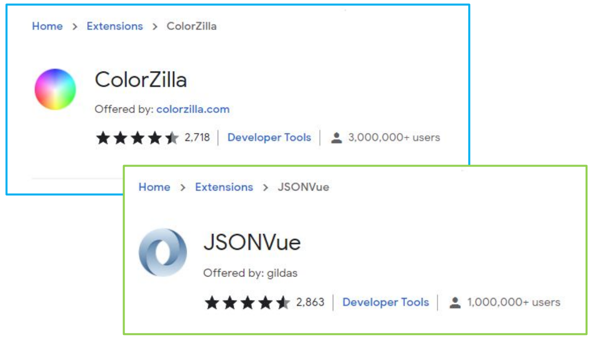
- Shortcut key: <*F12*>
- DevTools
  - *Console*
    - <*ESC*> toggles console
  - Elements
    - Select / Inspect Element <*CTRL+SHIFT+C*>
  - Toggle Device *(Mobile, etc.)*
    - *Responsive design*
  - Sources
  - Network
  - MORE…
    - Chrome Web Store
      - JSONVue
      - ColorZilla





DevTools - www.goo

```
<!DOCTYPE html>
<html itemscope it
 ▸<head>…</head>
```



Home › Extensions › ColorZilla

ColorZilla

Offered by: colorzilla.com

★★★★★ 2,718 | Developer Tools | 3,000,000+ users

Home › Extensions › JSONVue

JSONVue

Offered by: gildas

★★★★★ 2,863 | Developer Tools | 1,000,000+ users

Chrome DevTools *(Shortcuts)*

Opening DevTools

• Open last panel	⌘ Option I	F12 or Ctrl Shift I
• Open **Console** panel	⌘ Option J	Ctrl Shift J
• Open **Elements** panel	⌘ Shift C or ⌘ Option C	Ctrl Shift C

Global keyboard Shortcuts

• Toggle Device Mode	⌘ Option M	Ctrl Shift M
• Toggle Inspect Mode	⌘ Shift I	Ctrl Shift I
• Toggle Drawer	Escape	Escape
• Normal reload	⌘ R	Ctrl R or F5
• Hard reload	⌘ Shift R	Ctrl Shift R or Ctrl F5
• Zoom in	⌘ +	Ctrl +
• Zoom out	⌘ -	Ctrl -
• Restore default zoom	⌘ 0	Ctrl 0

Debugging / Sources Shortcuts

• Pause / Resume	⌘ \ or F8	Ctrl \ or F8
• Step Over	⌘ ' or F10	Ctrl ' or F10
• Step Into	⌘ + or F11	Ctrl + or F11
• Step Out	⌘ Shift + or Shift F11	Ctrl Shift + or Shift F11

Debugging

(1947, the first bug . . .)

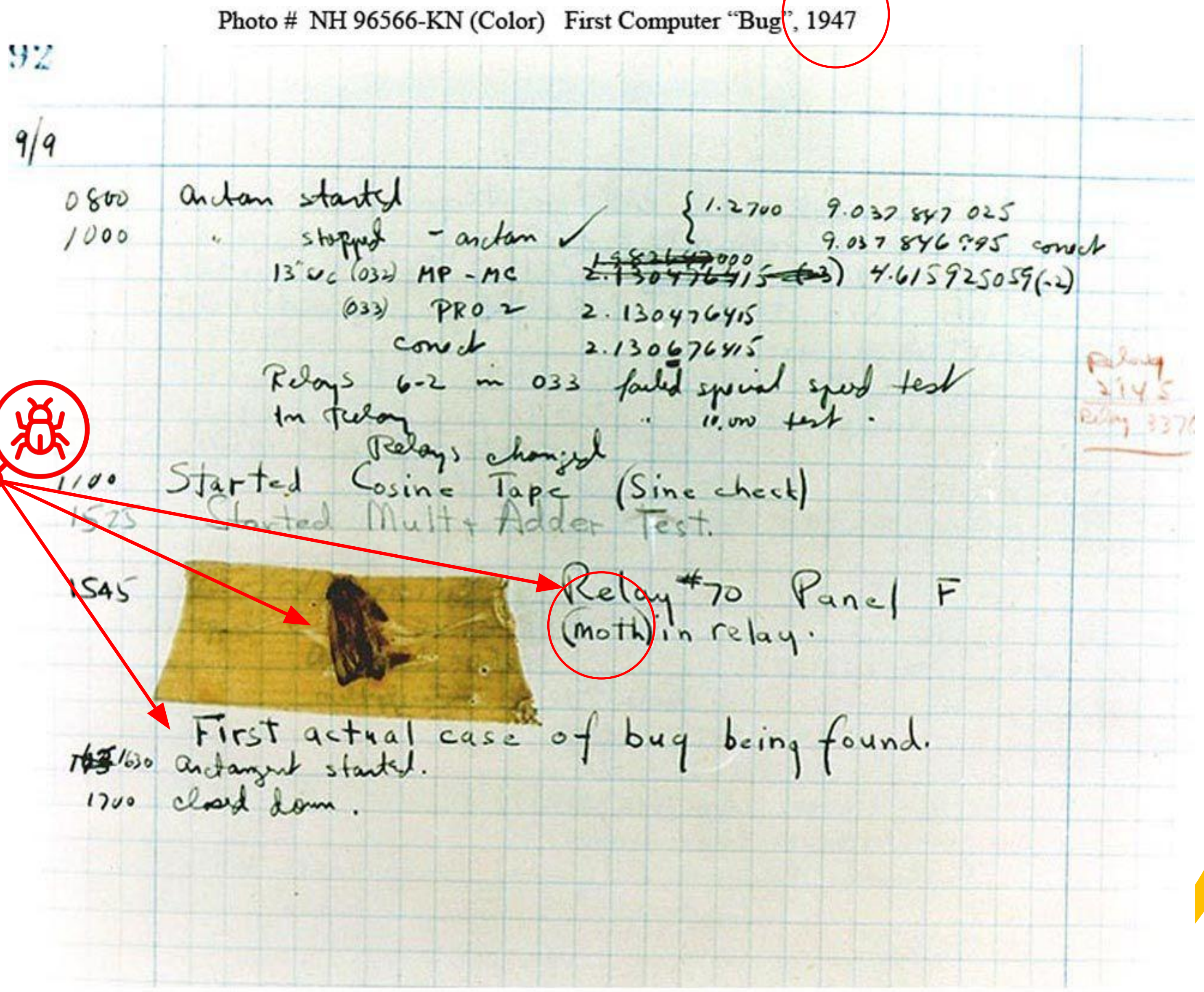


Photo # NH 96566-KN (Color) First Computer "Bug", 1947

92

9/9

0800 Anctan started

1000 " stopped - anctan ✓ {1.2700 9.037 847 025

9.037 846 995 conect

13" UC (032) MP - MC ~~1.982147000~~ ~~2.130476415 (-3)~~ 4.615925059(-2)

(033) PRO 2 2.130476415

conect 2.130676415

Relays 6-2 in 033 failed special speed test

In Relay " 10,000 test.

Relays changed

1100 Started Cosine Tape (Sine check)

1525 Started Mult + Adder Test.

1545 Relay #70 Panel F (moth) in relay.

First actual case of bug being found.

~~143~~ 1630 Antangent started.

1700 closed down.

Relay 2145

Relay 3370

Debugging

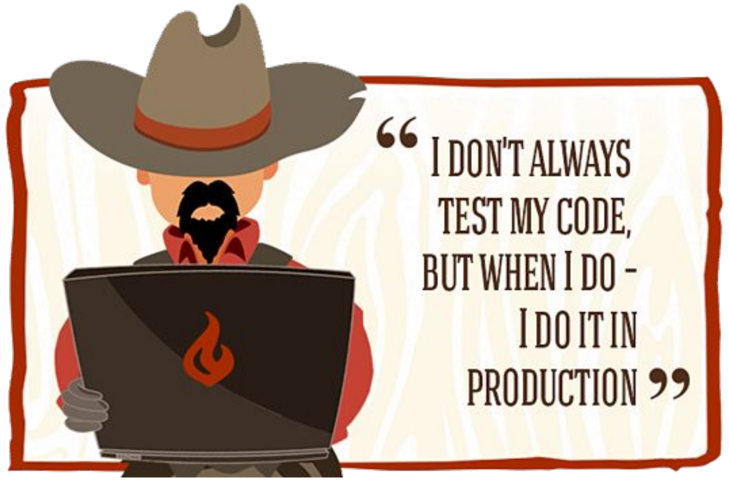
- Code runs… but doesn’t work as expected
  - 99% of time, the problem is "state" changed unintendedly
- Types
  - "**Poor Mans**" Debugger
    - `console.log()`
  - Integrated Debugging
    - Add breakpoint

Debugging / Sources Shortcuts

Pause / Resume	⌘ \ or F8	Ctrl \ or F8
Step Over	⌘ ' or F10	Ctrl ' or F10
Step Into	⌘ + or F11	Ctrl + or F11
Step Out	⌘ Shift + or Shift F11	Ctrl Shift + or Shift F11

Unit Testing

- Ensures that assumptions are true
  - Helps mitigate "ripple effect" in code where one change or fix breaks something else.
  - Test for Success **AND** Failure
- Code Coverage
- Test Driven Development (TDD)
  - Red => Green => Refactor => *(Repeat)*


"I DON'T ALWAYS TEST MY CODE, BUT WHEN I DO - I DO IT IN PRODUCTION"

Unit Testing
(continued)

- **AAA** Pattern *(TDD)*
  - **Arrange**
    Create variables, setup test case or scenario
  - **Act**
    Invoke condition or what you’re testing
  - **Assert**
    Validate that assumption or actions expected occurred.

- **GWT** Pattern *(BDD - Behavior)*
  - **Given**
    Describes the state of the world before you begin the behavior you're specifying in this scenario. You can think of it as the pre-conditions to the test.
  - **When**
    The behavior that you're specifying.
  - **Then**
    Describes the changes you expect due to the specified behavior.

Unit Tests

(mocha and chai)

- **Mocha**
  - Test framework for Node.js
- **Chai**
  - Assertion library for Node.js and browser
- NPM (Node Package Manager)
  - https://nodejs.org/en/download/
  - `npm init`
  - `npm install mocha chai --save-dev`
  - `npm test`

```
let expect = require("chai").expect;
function doSomething(x,y) {
  if (typeof x !== "string") throw new Error("x must be a string");
  return x + y;
}
describe("myFunctions", () => {
  describe("#doSomething", () => {
    it ("should concatenate", () => {
      let x = doSomething("Hello", 5);
      expect(x).to.equal("Hello5");
    });
    it ("should throw error if not string" , () => {
      expect(() => {
        doSomething(5, 5);
      }).to.throw(Error);
    });
  });
});
```

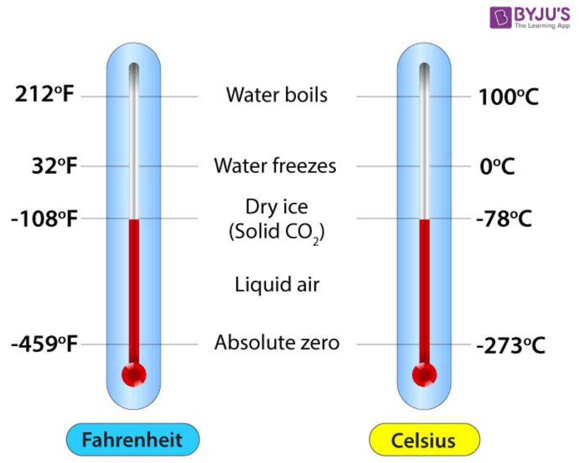
Test Driven Development

(Hands On)

`Temperature` Class

Requirements

- Create constructor that takes temperature in Fahrenheit
- Store the temperature as Fahrenheit inside the class
- The minimal accepted value for Fahrenheit is −459.67
  - Is there a maximum value?
- Create method to get value converted to Celsius
- Create method to get value converted to Kelvin value



DEMO

Test Driven Development / Unit Testing

Final Project

WAR!

- Classes
  - Card
  - Deck
  - Player
  - *Game?*
- Tests / Considerations (TDD)

  What happens when…
  - Shuffle empty deck?
  - Draw from empty deck?
  - Flip on empty hand?
  - One player runs out of cards?
  - Can a player have a negative score?
  - Describe when hand has no card?
- Requirements
  - Deal 26 Cards to two Players from a Deck.
  - Iterate through the turns where each Player plays a Card
  - The Player who played the higher card is awarded a point
  - A tie result in zero points for either Player
  - After all cards have been played, display the score.

Get in the habit of starting to think about **BAD** things and how to handle or prevent them.