CSE 115
Introduction to Computer Science I
Announcements

Finish module pre-lab *before* lab exam

For each lab exam you have a second chance, after the next lab exam.
Road map

▶︎ Review ◀

dependences/variables/assignment

functions

JavaScript on codenvy.io
Review

Expressions

Variables

Assignment

Defining functions: parameters, header, body

Calling functions: arguments, returned value, environments

Control flow: sequencing and selection (if/elif/else)
Road map

Review

▶ expressions/variables/assignment ⬅

functions

JavaScript on codenvy.io
JavaScript

expressions/variables/assignment

Simple expressions:

Literals (null, true, false, numeric literal, string literal)

All numbers are floating point.
JavaScript

expressions/variables/assignment

Compound expressions:

binary: expression operator expression

unary: operator expression or expression operator
Some binary operators:

- arithmetic: +, -, *, /, %, **
- string: +
- relational: <, <=, >, >=, ==, !=

Boolean (short circuiting): &&, ||
Some unary operators:

- arithmetic: +, -
- Boolean: !
Variables must be declared before use, and statements end with ';

```javascript
var x;

x = 13;

var y = 17;
```
Road map

Review

expressions/variables/assignment

▶ functions ◀

JavaScript on codenvy.io
JavaScript

defining and calling functions

Functions have same parts: header + body

def area(w, h):
    return w * h

function area(w, h) {
    return w * h;
}
defining and calling functions

keywords are different

def area(w, h):
    return w * h

function area(w, h) {
    return w * h;
}
JavaScript

defining and calling functions

Delimiters are different

def area(w, h):
    return w * h

function area(w, h) {
    return w * h;
}
Extra slide: this came up during class

Delimiter names

( ) are parentheses (singular: parenthesis)

[ ] are brackets

{ } are braces

The first of each pair is an opening or left delimiter, the second is a closing or right delimiter.
Extra slide:
this came up during class

Comments

```python
# This is a Python single-line comment
```

```javascript
// This is a JavaScript single-line comment
```

```javascript
/* This is a JavaScript comment that spans many lines. */
```
Extra slide: this came up during class

Additional Operators

= 
assignment

==
equality under type conversion ("loose" equality)

===
equality without type conversion ("strict" equality)

&&
logical AND

||
logical OR

!
logical NOT

&
bitwise AND

|
bitwise OR

~
bitwise NOT

We'll explain these operators (and what we mean by type conversion) later.
defining and calling functions

statement terminators needed in JavaScript

```python
def area(w, h):
    return w * h
```

```javascript
function area(w, h) {
    return w * h;
}
```

*While the language will allow semicolons to be omitted sometimes, it is safer to always insert them. This avoids subtle and difficult to track down bugs which can otherwise occur.*
JavaScript

defining and calling functions

Functions calls look similar

def area(w, h):
    return w * h

x = area(3, 7)

function area(w, h) {
    return w * h;
}

var x = area(3, 7);
Printing

print( 3 * 5 )

code

console.log( 3 * 5 );
JS

Python
Road map

Review

expressions/variables/assignment

functions

► JavaScript on codenvy.io ◄
To code along:

Create a workspace with 'node-default' stack

Create custom run command with:

`commandLine: cd ${current.project.path} && node hello.js`

`preview URL: http://${server.port.8000}`