Announcements

Important links now on UBLearns
Let’s talk about data!

What data is meaningful to you?

Any questions that came up?
A few examples of my own

Cleaning lists for fantasy football

01. Allen, Josh, QB, 17
02. Singeltary, Devin, RB, 26
03. Diggs, Stefon, WR, 14
04. Davis, Gabriel, WR, 13

Custom card generation

- Each suit goes from 1-16
- 16 different suits
- 256 cards in total

Turn this...

QB
Josh Allen
Justin Herbert

RB
Devin Singeltary
Dalvin Cook

WR
Stefon Diggs
Gabriel Davis
Cooper Kupp

...into these

hours to do by hand...

...python script takes minutes
Agenda

Basic principles for this course
Expressions
Demo: Expressions in Python
Basic Principles

Always keep in mind...change is inevitable (especially in tech)

Programming languages come and go

Core concepts remain

Programmers adapt and learn
Basic Principles

We aim to learn fundamental concepts common to all (or at least most) programming languages.

Specific details differ across languages.

I will try to differentiate between fundamental concepts and language specifics.

When in doubt, ask!

Many topics will be revisited/expanded upon.

Skills develop with practice.
Expressions, what are they?

An expression is a part of a program that has a value.
Expressions, what are they?

An expression is a part of a program that has a **value**

For example...

4  
3 + 7
Expressions, what are they?

An expression is a part of a program that has a value

for example...

4
Simple Expression

3 + 7
Compound Expression
Expressions, what are they?

An expression is a part of a program that has a value

*for example*...

Made up of multiple simple expressions...

4
Simple Expression

3 + 7
Compound Expression
Expressions, what are they?

An expression is a part of a program that has a value.

For example...

4
Simple Expression

3 + 7
Compound Expression

Can be expressed as a tree (this will be useful later)
Simple vs Compound

Simple Expressions
Atomic (cannot be decomposed)
Examples
12
2

Compound Expressions
Composed of multiple sub-expressions
Examples
2+7
-10 * 4 - 6
Simple Expressions (more than just numbers)

Types of simple expressions

- Numeric literals
  - `int` (ie: 2, 10, 465)
  - `float` (ie: 3.14, 17.999)

- Boolean literals
  - `bool` (True or False)

- Text literals (strings)
  - `str` (ie “Hello” or ‘goodbye’)
Simple Expressions (more than just numbers)

Types of simple expressions

- Numeric literals
  - int (ie: 2, 10, 465)
  - float (ie: 3.14, 17.999)
- Boolean literals
  - bool (True or False)
- Text literals (strings)
  - str (ie “Hello” or ‘goodbye’)

This syntax is specific to python
Compound Expressions

Compound expressions are made up of:

one or more expressions AND one operator

-17
[unary negation operator (-) with a simple expression (17)]

43 - 5
[binary subtraction operator (-), two simple expressions (43 and 5)]
What about more than one operator?

3 + 4 * 5, is it a valid expression?
What about more than one operator?

3 + 4 * 5, is it a valid expression?

Let’s break it down:
What about more than one operator?

3 + 4 * 5, is it a valid expression?

Let’s break it down:
What about more than one operator?

3 + 4 * 5, is it a valid expression?

Let’s break it down:
What about more than one operator?

3 + 4 * 5, is it a valid expression?

Let’s break it down:
What about more than one operator?

3 + 4 * 5, is it a valid expression?

Let’s break it down:
What about more than one operator?

3 + 4 * 5, is it a valid expression?

Let’s break it down:
Order of Operations

Follow rules from math:

Parenthesis first
Then multiplication/division
Then addition/subtraction

6 - 3 + (1 + 4 * 5) evaluates to 24

most programming languages follow this convention
Binary Operators (in Python)

Addition (+)
Subtraction (-)
Multiplication (*)
Division (/ or //)
Modulo (%)
Comparisons (<,<=,>,>=,==,!=)
Binary Operators (in Python)

Addition (+)
Subtraction (-)
Multiplication (*)
Division (/ or \)
Modulo (%)
Comparisons (<, <=, >, >=, ==, !=)

/ and // give fractional and integer results respectively.
% calculates the remainder.
10 / 3 evaluates to 3.333333
10 // 3 evaluates to 3
10 % 3 evaluates to 1
String Expressions

Expressions are evaluated to produce their values

The expression “hello” has value “hello”

The expression “hello “ + “world” is “hello world”

Note the space in “hello “
String Expressions

Expressions are *evaluated* to produce their values.

The expression “hello” has value “hello”.

The expression “hello “ + “world” is “hello world”.

*Note the space in “hello “*

Here (+) is the string concatenation operator...context matters!

*we also saw this with unary (-) vs binary (-)*
Demo in Replit