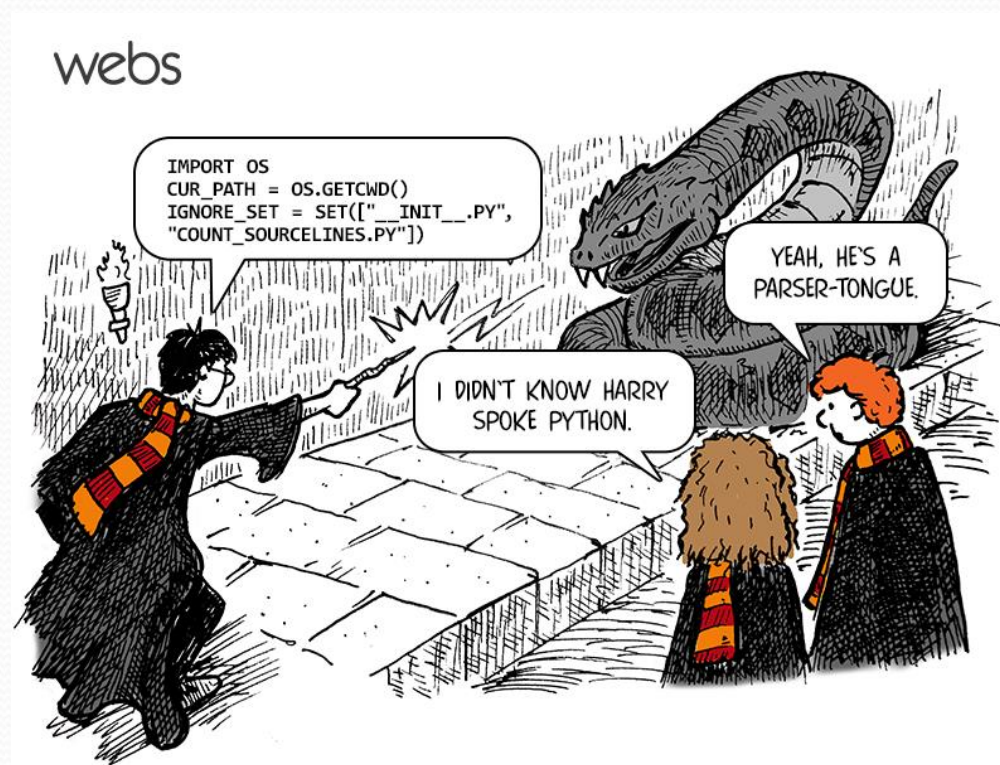


CS 115, Autumn 2021

Lecture 5: variables; expressions



Thanks to Marty Stepp and Stuart Reges for parts of these slides

Variable

- **variable declaration and assignment:**

Sets aside memory for storing a value and stores a value into a variable.

- Syntax:

name = expression

- `name = "Merlin"`

- `last_name = "Cat"`

name	"Merlin"
------	----------

last_name	"Cat"
-----------	-------

- **variable usage:**

Once given a value, a variable can be used anywhere its value can be used:

- `print(name)`

Arithmetic operators

- **operator**: Combines multiple values or expressions.

+ addition

- subtraction (or negation)

* multiplication

/ division

// integer division (a.k.a. leave off any remainder)

% modulus (a.k.a. remainder)

** exponent

- As a program runs, its expressions are *evaluated*.
 - $1 + 1$ evaluates to 2

Expressions

- **expression:** A value or operation that computes a value.
 - Examples:
 $1 + 4 * 5$
 $(7 + 2) * 6 / 3$
 42.0
 - The simplest expression is a *literal value*.
 - A complex expression can use operators and parentheses.

Variables and expressions

- Once given a value, a variable can be used in expressions:

```
x = 3          # x is 3
y = 5 * x - 1  # now y is 14
```

- You can assign a value more than once:

```
x = 3          # 3 here
```

x	11
---	----

```
x = 4 + 7      # now x is 11
```

Assignment and algebra

- Assignment uses $=$, but it is not an algebraic equation.
 - $=$ means, *"store the value at right in variable at left"*
 - The right side expression is evaluated first, and then its result is stored in the variable at left.
- What happens here?

$x = 3$

$x = x + 2$ # ???

x	5
---	---

Exercise: Grade calculator

- Write a program that prompts a user for two project scores, points possible and the weight of the projects in the overall grade and outputs:
 - the total points earned
 - the total points possible
 - the total percentage
 - the total grade points earned

Exercise: Grade calculator

- Example output:

What is the first project score? 9

How many points are possible on the first project? 10

What is the second project score? 15

How many points are possible on the second project? 20

What percentage of the grade are projects worth? 45

Total points earned: 24

Total points possible: 30

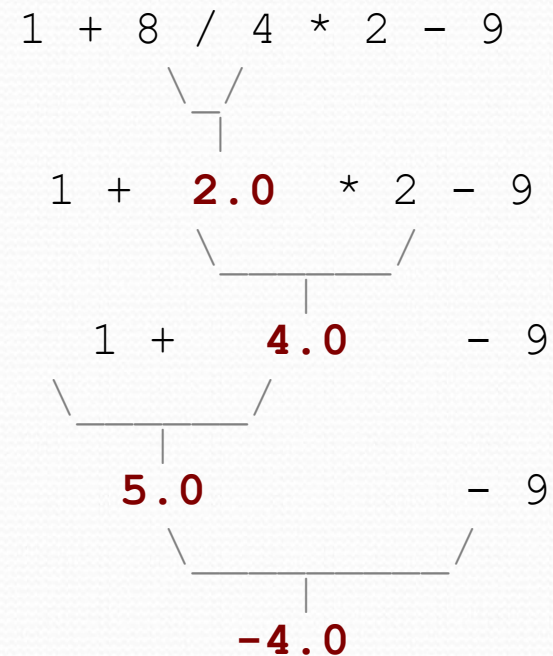
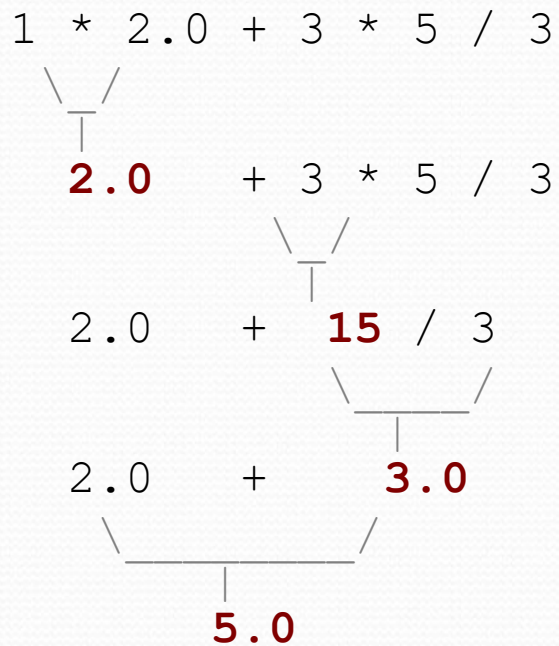
Total percentage earned: $24/30 = 0.8 = 80\%$

Total grade points earned: 36

Precedence

- **precedence:** Order in which operators are evaluated.
 - Generally operators evaluate left-to-right.
 $1 - 2 - 3$ is $(1 - 2) - 3$ which is -4
 - But $*$ / have a higher level of precedence than $+$ -
 $1 + 3 * 4$ is 13
 $6 + 8 / 2 * 3$
 $6 + 4.0 * 3$
 $6 + 12.0$ is 18
 - Parentheses can force a certain order of evaluation:
 $(1 + 3) * 4$ is 16
 - Spacing does not affect order of evaluation
 $1+3 * 4-2$ is 11

Precedence examples



Integer division with //

- When we divide integers with //, the quotient is also an integer.

- $14 // 4$ is 3, not 3.5

$$\begin{array}{r} \underline{\quad 3} \\ 4 \) \ 14 \\ \underline{12} \\ 2 \end{array}$$

$$\begin{array}{r} \underline{\quad 4} \\ 10 \) \ 45 \\ \underline{40} \\ 5 \end{array}$$

$$\begin{array}{r} \underline{\quad 52} \\ 27 \) \ 1425 \\ \underline{135} \\ 75 \\ \underline{54} \\ 21 \end{array}$$

- More examples:

- $32 // 5$ is 6
- $84 // 10$ is 8
- $156 // 100$ is 1

- Dividing by 0 causes an error when your program runs.

Precedence questions

- What values result from the following expressions?
 - $9 // 5$
 - $695 \% 20$
 - $7 + 6 * 5$
 - $7 * 6 + 5$
 - $248 \% 100 / 5$
 - $6 * 3 - 9 // 4$
 - $(5 - 7) * 2 ** 2$
 - $6 + (18 \% (17 - 12))$