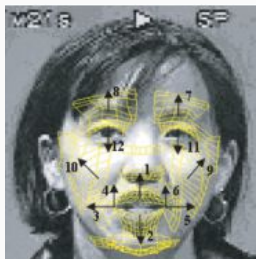


Welcome!

Making Game with Python (1)

Zhihong (John) Zeng & Andrew Zeng

Zhihong (John) Zeng

[illegible]

Andrew Zeng



Andrew Jiashu Zeng's Homepage

[HOME](#)[ANIMALS](#)[GAMES](#)[BLOG](#)[CONTACT](#)

My homepage has some of my ideas, so you can explore this website.

I love animals.

Here is my 4-H visual presentation about OWL.



Today

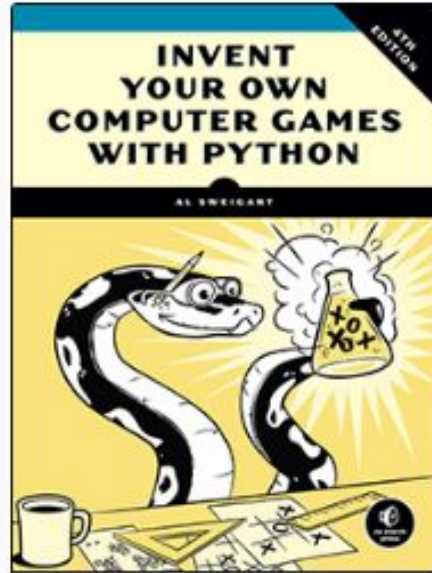
- School Rule
- Course info
- Computer fundamentals
- Python basics
- Mathematical operations
- Python variables and types
- Exercise

School Rule

1. Students must address teachers and parents respectfully at all times.
2. Students must attend the class(es) on time with all the needed and required supplies, materials, and instruments.
3. Students must wait at assigned area if coming in early.
4. Students must follow classroom rules for appropriate behavior.
5. No running, chasing, and/or yelling in corridors when class is in recess.
6. Students must respect and take good care of school property.
7. Students must help keep classroom clean and neat. No food or drinks are allowed in classroom.

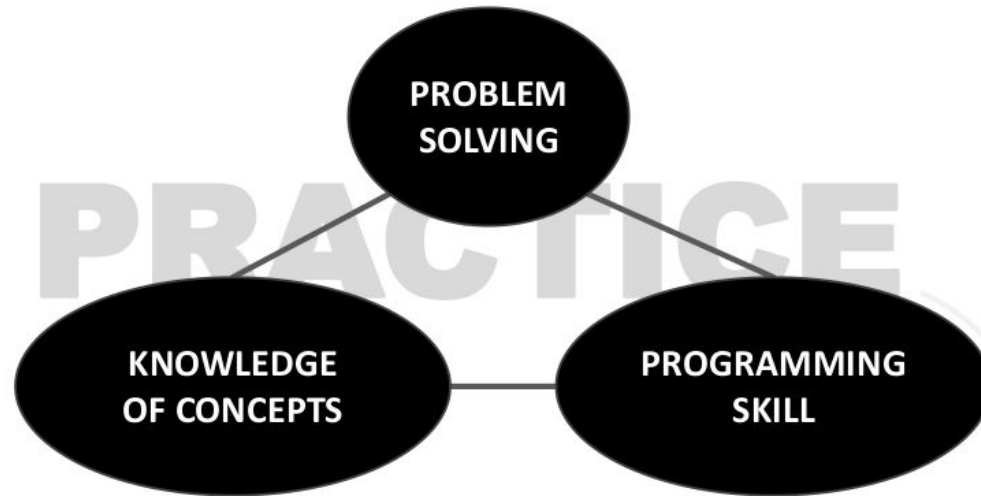
Safety: If you cannot find your parents after class, please notify your teacher or school staff at Room 409

Course Info

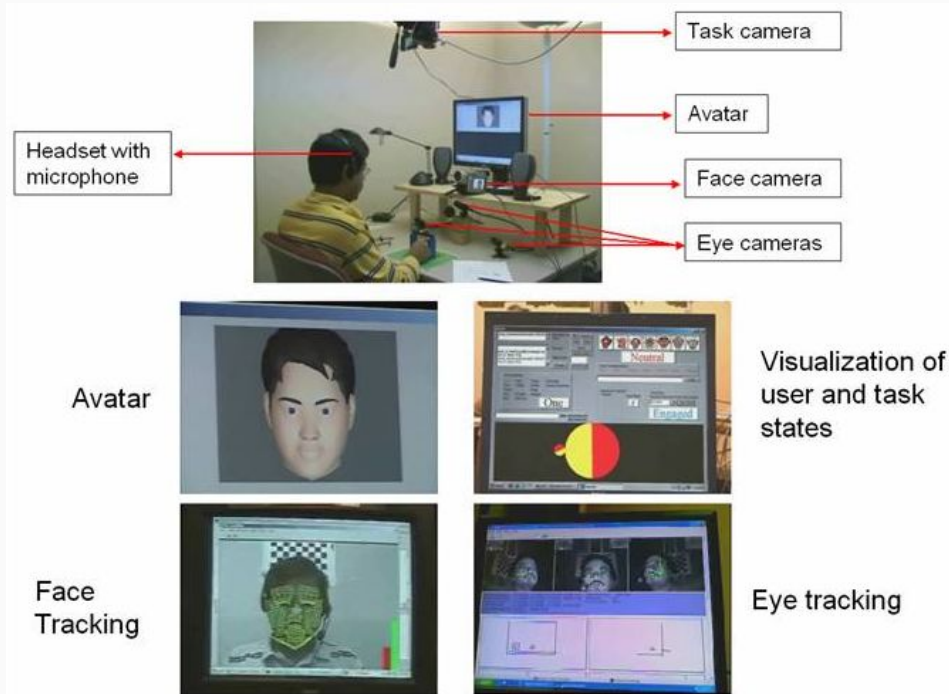


<http://inventwithpython.com/invent4thed/>

Course Info



Computer Game: Human-computer Interaction

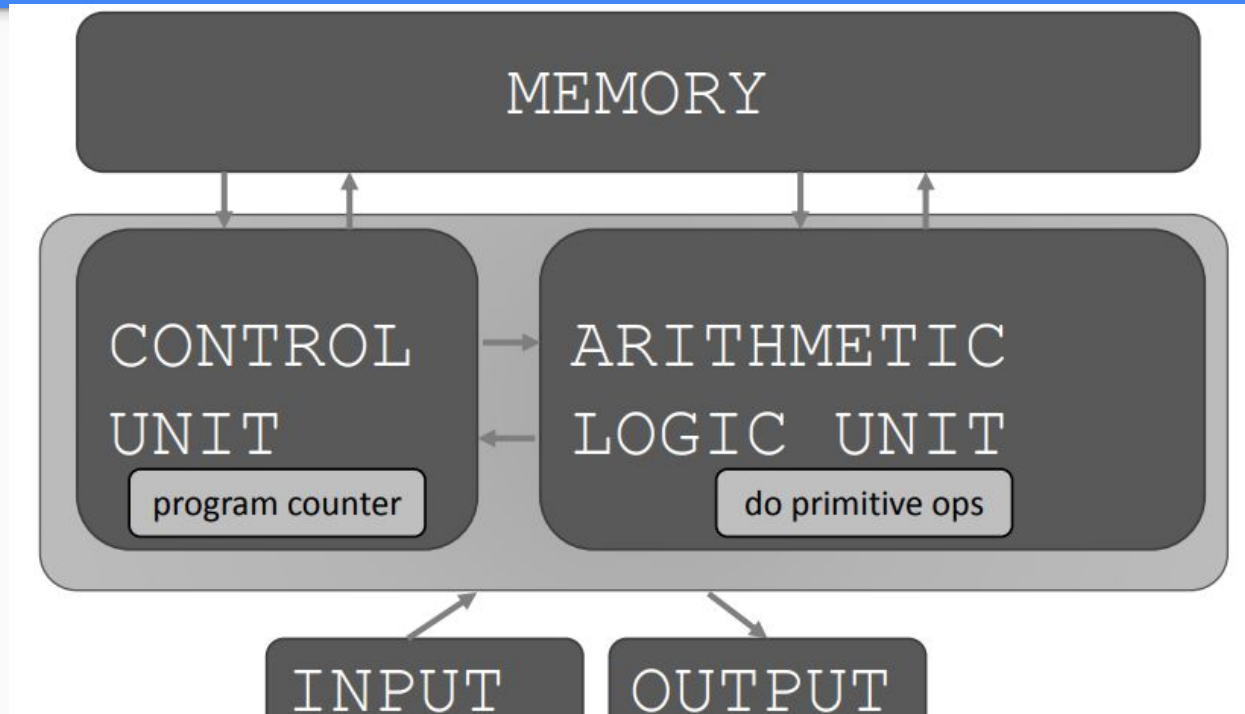


Computer Fundamentals

What does a Computer do?

- Fundamentally:
 - Performs calculations
 - Remembers results
- What kinds of calculations:
 - Built-in to the language
 - Ones that you define as the programmer
- Computers only know what you tell them

Basic Computer Architecture



What is a programming recipe

- Sequence of simple steps
- Flow of control process that specifies when each step is executed
- A means of determining when to stop

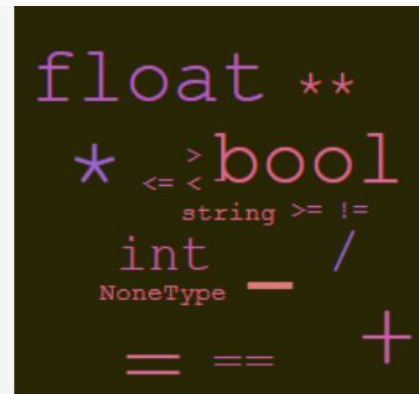
`1+2+3 = an algorithm`

Aspects of Languages

- Primitive constructs
 - English: words
 - Programming language: numbers, strings, simple operators



Word Cloud copyright [Michael Twardos](#), All Right Reserved. This content is excluded from our Creative Commons license. For more information, see <https://ocw.mit.edu/help/faq-fair-use/>.



Word Cloud copyright unknown, All Right Reserved.
This content is excluded from our Creative Commons license. For more information, see <https://ocw.mit.edu/help/faq-fair-use/>.

Aspects of Languages

- Syntax
 - English:
 - “Cat dog boy” -> not syntactically valid
 - “Cat hugs boy” -> syntactically valid but
 - Programming language
 - “hi” 5 -> not syntactically valid
 - $3.2 * 5$ -> syntactically valid

Aspects of Languages

- Language meaning
 - English: Can have many meaning
 - “Flying planes can be dangerous”
 - Programming language: can have only one meaning but may not be what a programmer intend

Python Basics

Python Programs

- A program is a sequence of definitions and commands
 - Definitions evaluated
 - Commands executed by Python interpreter in a shell
- Commands (statements) instruct interpreter to do something
- Can be typed directly in a shell or stored in a file that is read into the shell and evaluated intended

Scalar Objects

- `int` -- represent integers, ex. 5
- `float` -- represent real numbers, ex. 3.27
- `bool` -- represent Boolean values True and False
- `NoneType` -- special and has no value, None
- Can use `type()` to see the type of an object
 - `>>>type(5)` -->int
 - `>>>type(3.0)` -->float

Type Conversions (Cast)

- Can convert object of one type to another
 - `float(3)` converts integer 3 to float 3.0
 - `int(3.9)` truncates float 3.9 to integer 3
 - `int('321')` converts string '321' to integer 321
 - `str(123)` converts integer 123 to string '123'

Printing to Console

- To show output from code to a user, use print command
 - `print(3)`
 - `print(3+2)`
 - `print('ABC')`

Expressions

- Combine objects and operators to form expressions
- An expression has a value, which has a type
- Syntax for a simple expression
 - `<object> <operator> <object>`

Operators on ints and floats

- $i + j \rightarrow \text{sum}$
- $i - j \rightarrow \text{difference}$
- $i * j \rightarrow \text{product}$

Note: if both are ints, result is int. If either or both are floats, result is float

- $i / j \rightarrow \text{division. result is float (python3) or depends (python2)}$
- $i \% j \rightarrow \text{remainder when } i \text{ is divided by } j$
- $i ** j \rightarrow i \text{ to the power of } j$

Operation precedence (order)

- Parentheses used to tell Python to do these operations first
- Operator precedence without parentheses
 - `**`
 - `*`
 - `/`
 - `+` and `-` - executed left to right, as appear in expression

Binding Variables and Values

- Equal sign is an assignment of a value to a variable name
 - $\text{Pi} = 3.14$
 - $\text{Pi_approx} = 22/7$
- Value stored in computer memory
- An assignment binds name to value
- Retrieve value associated with name or variable by invoking the name

Abstracting Expressions

- Why give names to values of expressions
 - To reuse names instead of values
 - Easier to change code later

```
pi = 3.14  
radius = 2.2  
area = pi * (radius ** 2)
```

Programming vs Math

- In programming, you do not “solve for x”
- Programming:
 - Expression on the right
 - Variable name on the left
 - Bonus: Equivalent expression to $\text{radius} = \text{radius} + 1$ is $\text{radius} += 1$

Python:

```
pi = 3.14  
radius = 2.2  
area = pi * (radius ** 2)
```

Math:

```
pi = 3.14  
radius = 2.2  
pi * (radius ** 2) = area
```

Open computer -> run python

