Lecture 1

Course Introduction and Basic Operators

Scheduling

Lecture Hours

-Tuesday/Thursday 4:30 - 6pm PST

Hosted through zoom and lectures will be recorded for people who miss it. Try your best to attend it live.

Discussion Hours

-Wednesday 5:30 - 7:30pm

You will all be able to discuss the topics as well as your assignments on the discord at any time but during this time the TA and I will be active as well to help

Office Hours

-Mine: Fridays anytime

-Brian: Mondays anytime

You can come to us during these days for one on one help with assignments

Why Python?

Pros:

-Dynamically typed

-Extensive Libraries

-Easy to learn and use

-Readability

-Data science

Cons:

-Slow

-Not good for mobile computing and browser applications

-Less strict coding standard

Course Goals

My Goals:

-Grow and develop my skills as an instructor

-Aid everyone on the things that I struggled the most with when I was in their position

Your Goals:

-Build a strong foundation that will aid you in upcoming classes as an ICS major

-See personal development as a programmer

Course Etiquette

I have worked extremely hard these past months developing this course for you all in hopes that everyone here will be able to use the skills that we teach to develop their own in order to become successful in their future classes.

We have tried our best to replicate a remote university class setting but the only thing that we can not replicate is the incentive of a GPA so the only true incentive we can give you is the desire to learn.

TLDR: Don’t procrastinate, do the projects and assignments on time, don’t cheat, if you need help ask for help, etc. We are here to help you not to give you a value.

What will we be covering?

All of ICS 31 Topics

- Basic syntax and functions
- Basic data types and structures
- Control structure
  - If/elif/else
  - while/for
  - Function definition/calling/passing
- Operations
  - Arithmetic
  - Comparison
  - Logical
- File handling
- Basic algorithms
- Basic built in functions

ICS 32 & 33 Topics (if there is time)

- Classes
- Recursion
- Comprehensions
- External Libraries and Modules
- Good practices
- Proper Debugging
- Error handling

Course Assignments

Projects:

- Assigned 1 - 3 days a week depending on difficulty
  - Assignments will start out small until we start learning more features around the beginning of week 2
- Projects will be due the night before lectures so we can go over the solutions together
- Project write ups can be found in the class's google drive

Weekly Review Assignments:

- These assignments will be assigned once a week and will consist of smaller problems focused on using the skills taught during the week
- Will consist of problems such as fill in the blank/mc/etc.
- These will be due on Tues before lecture
- You can also find these in the class's google drive

What is Programming

Learning how to program is like learning a new language.

- There are many ways of saying the same message but some ways are better than others
- You can be taught what the words mean but it takes practice to be able to use/speak it properly
- Assuming you have some knowledge of basic words I can tell you what it means and does and you'll be able to understand it but the hard part is coming up with it on your own

Just like learning a new language, learning to program requires practice. Programming is different from a speaking language as it is based around logical statements.

Programming in General

- **The program is going to take input(s)**
  - Ex: files, keyboard, network, etc.
- **The program will process the input(s)**
  - Ex: adding 2 values, etc
- **The program will output the processed data**
  - Ex: files, screen, network, etc.

Programming is like a recipe. For example, I want to make pasta.

I will first input pasta, water, sauce, etc.

Then I will cook and combine everything

Finally I will output the finished product

What is the “Standard Library”

In python there exists what is called the Standard Library which contains all the basic functions and objects and is built in to the language itself. As a result we can call on these functions and objects without having to import or define them.

Basic Data Types & Object Assignment

- String
  - “Hello”, ‘My name is Trung’
- Integers
  - 1, 4, 123
- Float
  - 1.32, 1.0, 1249.123
- Boolean
  - True / False

You can identify an object’s type by using the function type()

Object Assignment

General Format

variable = assigned object

Examples:

name = ”trung”

name = trung

age= “19”

age = 19

What does it mean to be dynamically typed?

Unlike Java and C++ are statically typed languages where you would declare a variable with its type and for the life of that variable it can not change types:

C++:

int variable = 4;

Python on the other hand is a dynamically typed language which means variables can change type at any time and its type doesn’t need to be stated at runtime.

GIVE YOUR VARIABLES MEANINGFUL NAMES!

Give your variables names that makes sense for what you are currently doing with them!

Example: I want a variable to keep track of my calorie intake

Good:

myCalories = 100

Bad:

Haha = 100

x = 100

Basic Operators

Relational:

(>) Greater Than

(<) Lesser Than

(==) Equal to

(!=) Not Equal to

(>=) Greater Than or Equal To

(<=) Less than or Equal To

Logical:

and

or

not

Arithmetic:

(+) Addition

(-) Subtraction

(*) Multiplication

(/) Division

(**) Exponentiation

(%) Modulus

(//) Floor Division

Compound

(+=) Plus Equals

(-=) Minus Equals

(*=) Multiplication Equals

(/=) Division Equals

(%=) Modulus Equals

Difference between = and ==

You would use = when you are assigning a value to a variable and == when you want to check if two variables are equal.

When you use = this means an assignment.

- x = 4
  - What this means is that the variable x is assigned to the variable int 4

When you use == this is used to see if one object is equal to another object

- x = 4
- y = 4
- x==y

Basic Operators Example

Arithmetic Examples:

- **(+)** Addition:
  - x = 5 + 3
  - x == 8
- (-) Subtraction:
  - x = 2 - 2
  - x == 0
- (*) Multiplication:
  - x = 3 * 2
  - x == 6
- (/) Division:
  - x = 4 / 2
  - x == 2
- (**) Exponentiation:
  - x = 3 ** 3
  - x == 27
- (%) Modulus (remainder):
  - x = 3 % 2
  - x == 1
- (//) Floor Division (rounds down to nearest whole number):
  - x = 3 // 2
  - x == 1

Updating Variables

How do we update variables?

- One way is to use compound operators but another way is doing it in a more explicit assignment statement

ex)

- x = 2
- x = x + 2
- x == 4

What happened in the second bullet point is saying that x is equal to x (which is 2 from the previous statement) + 2 which results in 4

Basic Operators Example

Compound Operators:

- **(+=)** Plus Equals:
  - x = 5
  - x += 3
  - x == 8
- (-=) Minus Equals:
  - x = 2
  - x -= 1
  - x == 1
- (*=) Multiplication Equals:
  - x = 3
  - x *= 2
  - x == 6
- (/=) Division Equals:
  - x = 4
  - x /= 2
  - x == 2
- (%=) Modulus Equals:
  - x = 3
  - x %= 2
  - x == 1

Basic Functions

print()

Use the print function to print objects to the console

```
>>> print("Hello Boo")
Hello Boo
```

input()

The input function will print a string to the console and waits for a user input.

The user input is then taken in as a string and assigned to the variable.

In this case the variable *fun* will be “A lot”

```
>>> fun = input("How much fun are you having")
How much fun are you having>? A Lot
```

Implicit Type Conversion

As stated in the last slide when you call input() and assign it to a variable the variable's type is String object.

But what if you wanted the inputted variable to be something other than a string?

Note: Implicit type conversion can be used in many scenarios that you will encounter throughout your programming career. Input is the the most basic example

```
1  variableA = input("Variable A: ")
2  print(variableA)
3  print(type(variableA))
4  variableB = int(input("Variable B: "))
5  print(variableB)
6  print(type(variableB))
```

Output:

Implicit Type Conversion

As stated in the last slide when you call input() and assign it to a variable the variable's type is String object.

But what if you wanted the inputted variable to be something other than a string?

Note: Implicit type conversion can be used in many scenarios that you will encounter throughout your programming career. Input is the the most basic example

```
1  variableA = input("Variable A: ")
2  print(variableA)
3  print(type(variableA))
4  variableB = int(input("Variable B: "))
5  print(variableB)
6  print(type(variableB))
```

Output:

```
4
<class 'str'>
Variable B: 4
4
<class 'int'>
```

Try it yourself: Inputs & Print

Write a statement that asks the user for their favorite food and assign it to a variable called food then print the food variable.

Try it yourself: Inputs & Print

Write a statement that asks the user for their favorite food and assign it to a variable called food then print the food variable.

```
1  food = input("What is your favorite food? ")
2  print(food)
```

Output:

```
What is your favorite food? pizza
pizza
```

How do we use implicit type conversions?

We can use implicit type conversions by calling the type that we want to convert it to on the object.

Integers - int()

Float - float()

String - str()

Valid Examples:

- int(‘5’) -> 5
- int(42.129) -> 42
- float(“5”) -> 5.0
- float(32) -> 32.0
- str(5) -> “5”
- str(4.2) -> “4.2”

Non-valid Example

- int(“hello”)
- float(True)

Try it yourself: Implicit type conversions

What will these output:

Output:

str(“5.0”)

int(“My name is Brian”)

float(int(5.23))

str(float(3.023))

int(4)

Try it yourself: Implicit type conversions

What will these output:	Output:
str(“5.0”)	“5.0”
int(“My name is Brian”)	Error
float(int(5.23))	5.0
str(float(3.023))	“3.023”
int(4)	4

Conditional Statements

if statements are used to execute a command if a certain condition is met.

else statements execute a command if the original condition is not met.

```
x=5
y=3
if(x<y):
    print("Hi!")
else:
    print("Bye!")
```

Output:

if/elif statements are used to check multiple conditions in order.

Note: once one of the conditions is satisfied, none of the other conditions are checked.

```
score= 92

if (score>90):
    print("A")
elif (score>80):
    print("B")
elif (score>70):
    print("C")
else:
    print("You did not pass")
```

Output:

Conditional Statements

if statements are used to execute a command if a certain condition is met.

else statements execute a command if the original condition is not met.

```
x=5
y=3
if(x<y):
    print("Hi!")
else:
    print("Bye!")
```

Output:

“Bye!”

if/elif statements are used to check multiple conditions in order.

Note: once one of the conditions is satisfied, none of the other conditions are checked.

```
score= 92

if (score>90):
    print("A")
elif (score>80):
    print("B")
elif (score>70):
    print("C")
else:
    print("You did not pass")
```

Output:

“A”

Conditionals Cont.

Understanding the difference between if/elif/else is very important in being able to control the structure of the conditional statement.

Example:

What do you think this will print?

```
1  x = 10
2
3  if x < 20:
4      print("a")
5  if x < 30:
6      print("b")
7  if x < 40:
8      print("c")
9  elif x < 50:
10     print("d")
11 else:
12     print("e")
```

Conditionals Cont.

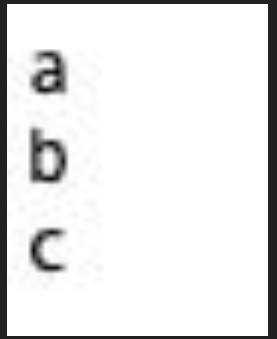
Understanding the difference between if/elif/else is very important in being able to control the structure of the conditional statement.

Example:

What do you think this will print?

```
1  x = 10
2
3  if x < 20:
4      print("a")
5  if x < 30:
6      print("b")
7  if x < 40:
8      print("c")
9  elif x < 50:
10     print("d")
11 else:
12     print("e")
```

Output:



Try it yourself: If/elif/else Conditionals

Write a script that can determine what grade an user inputted number will be.

A : >=90

B: >=80

C: >=70

Fail: <=60

Solution:

Try it yourself: If/elif/else Conditionals

Write a script that can determine what grade an user inputted number will be.

A : >=90

B: >=80

C: >=70

Fail: <=60

Solution:

```
grade = float(input("What is your grade? "))

if grade >= 90:
    print("A")
elif grade >= 80:
    print("B")
elif grade >=70:
    print("C")
else:
    print("failed")
```

```
grade = float(input("What is your grade? "))

if grade >= 90 and grade <= 100:
    print("A")
if grade >= 80 and grade < 90:
    print("B")
if grade >= 70 and grade < 80:
    print("C")
else:
    print("failed")
```