

For most, if not all of this class, we'll be working in IDLE, the simple development environment that comes with python. You can get it for free, with the Python language, at [python.org](http://python.org).

When you open IDLE, you're in a python shell. Type a single line and hit enter. The interpreter will evaluate, and sometimes print some output.

Try basic arithmetic expressions. Learn to use the following operators: `+` `-` `*` `/` `**` `//` `%`.

```
> 4 + 2
```

```
> 4 / 2.3
```

```
> 4 % 3
```

```
> 5 > 3
```

```
> 6 % 3
```

```
> 7 % 3
```

```
> 7 // 3
```

```
> 3**0
```

```
> 3**1
```

```
> 3**2
```

Next, try setting some variables.

```
> x = 3
```

```
> y = 2.1
```

```
> x
```

```
3
```

```
> x + y * y
```

```
7.41
```

Now try manipulating variables, resetting them as you do calculations.

```
> x = x + 3
> x += 3
> y = 3*x + x**1.5
> y /= 2
```

Let's do more with text.

```
> x = 'A well regulated militia being necessary for the defense of a free state'
> x
```

```
'A well regulated militia being necessary for the defense of a free state'
```

The data in variable `x` is a string, meaning a string of characters. Python supports some operations on strings that make it easy to create messages. Don't try this in C++. It won't work there.

```
> y = ', the right of the people to keep and bare arms'
> x + y
```

A well regulated militia being necessary for the defense of a free state, the right of the people to keep and bare arms

What happens if we need a number in a string?

```
> am = 'Ammendment '
> am + 2
```

Error...

How can we fix that?

Type Conversions...

```
> am + str(2)
```

The conversions we're going to look at are `str()`, which returns a string, `int()`, which returns a signed integer, and `float()`, which returns a floating point decimal. Try them with arbitrary values.

Finally, let's use the `input()` function to get some input from the user.

```
> usersays = input("type a prompt here")
```

Now, let's leave the shell and write some scripts. Python scripts are programs that run one line at a time at a time, through an entire file. Create a new file and save it to the desktop. We'll look at example code from page 21 of *Automate the Boring Stuff With Python* by Al Sweigart, and at some code that I wrote.

```
# This program says hello and asks for my name.  
# Code taken from page 21 of Automate the Boring Stuff With Python,  
# by Al Sweigart, No Starch Press. Al code below stars appears in book.  
  
# *****  
  
# This program says hello and asks for my name.  
  
print('Hello world!')  
print('What is your name?')          # ask for their name  
myName = input()  
print('It is good to meet you' + myName)  
print('The length of your name is:')  
print(len(myName))  
print('What is your age?')          # ask for their age  
myAge = input()  
print('You will be ' + str(int(myAge) + 1) + ' in a year')
```

Next Program is from me. It contains some control flow, which we'll look at next.

# Written by Shmuel Jacobs to simulate St. Petersburg lottery problem.

```
from random import randint as randint
```

# Variable holds accumulated prize money from all runs.

```
total = 0
```

# Constant controls how many times game is played.

```
NUMRUNS = 1000000
```

```
hiscore = 0
```

```
for x in range(NUMRUNS):
```

```
    # Flip coin.
```

```
    lastflip = randint(1,2)
```

```
    # Coin has been flipped once, so prize has doubled 0 times.
```

```
    num_doubles = 0
```

```
    # Flip coin until getting 'tails', recording number of flips.
```

```
    while(lastflip == 1):
```

```
        num_doubles+=1
```

```
        lastflip = randint(1,2)
```

```
    # Calculate prize.
```

```
    prize = 2**num_doubles
```

```
    # Add to accumulated total.
```

```
    total += prize
```

```
    # Update High Score
```

```
    if prize > hiscore:
```

```
        hiscore=prize
```

```
    # print("Just won", prize)
```

```
# Calculate and print average winnings.
```

```
print("Mu converges at ", total/NUMRUNS, ".")
```

```
print("Highest prize was ", hiscore)
```