



# Computer Programming Club #1

PJ Gibson

# Today's Agenda

- 1) Learn to store information**
- 2) Float, Integer, String**
  - a) Learn to differentiate
  - b) Learn rules of operators
  - c) Learn to convert
- 3) Learn about user input**
- 4) Learn how to print information**
- 5) Solve Challenges**

# Storing Information



**Rule:** You can store any information by giving it a variable

**Limitations** → You **can not** start a variable name with a number  
You **can not** use spaces in a variable name  
(use \_ instead)

## Good (will work)

```
In [13]: ▮ gravity = 9.8
```

```
In [14]: ▮ gravity*2
```

```
Out[14]: 19.6
```

```
In [16]: ▮ acceleration_due_to_gravity = 9.8
```

```
In [17]: ▮ acceleration_due_to_gravity*2
```

```
Out[17]: 19.6
```

## Bad

```
▮ 1gravity = 9.8
```

```
File "<ipython-input-8-16212c375d1f>", line 1  
  1gravity = 9.8  
    ^
```

```
SyntaxError: invalid syntax
```

```
▮ acceleration due to gravity = 9.8
```

```
File "<ipython-input-9-65f21dfbf30e>", line 1  
  acceleration due to gravity = 9.8  
    ^
```

```
SyntaxError: invalid syntax
```

# Storing Information

Example: Avogadro's Number  $L = 6.02 \times 10^{23}$

Number of  
Moles

×

Avogadro's  
Number

=

Number of  
Particles

You want to find number of particles in **5**, **23**, **303**, and **4000** moles

```
part1 = (6.02 * 10**23)*5
part2 = (6.02 * 10**23)*23
part3 = (6.02 * 10**23)*303
part4 = (6.02 * 10**23)*4000
```

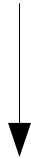
=

```
L = 6.02 * 10**23
part1 = L*5
part2 = L*23
part3 = L*303
part4 = L*4000
```

=

```
baba_lao = 6.02 * 10**23
part1 = baba_lao*5
part2 = baba_lao*23
part3 = baba_lao*303
part4 = baba_lao*4000
```

```
part1 = (6.02 * 10**23)*5
part2 = (6.02 * 10**23)*23
part3 = (6.02 * 10**23)*303
part4 = (6.02 * 10**23)*4000
```



Lots of Typing  
A calculator can do this

**Bad**


```
L = 6.02 * 10**23

part1 = L*5
part2 = L*23
part3 = L*303
part4 = L*4000
```



Looks nice  
Little typing

**Best**



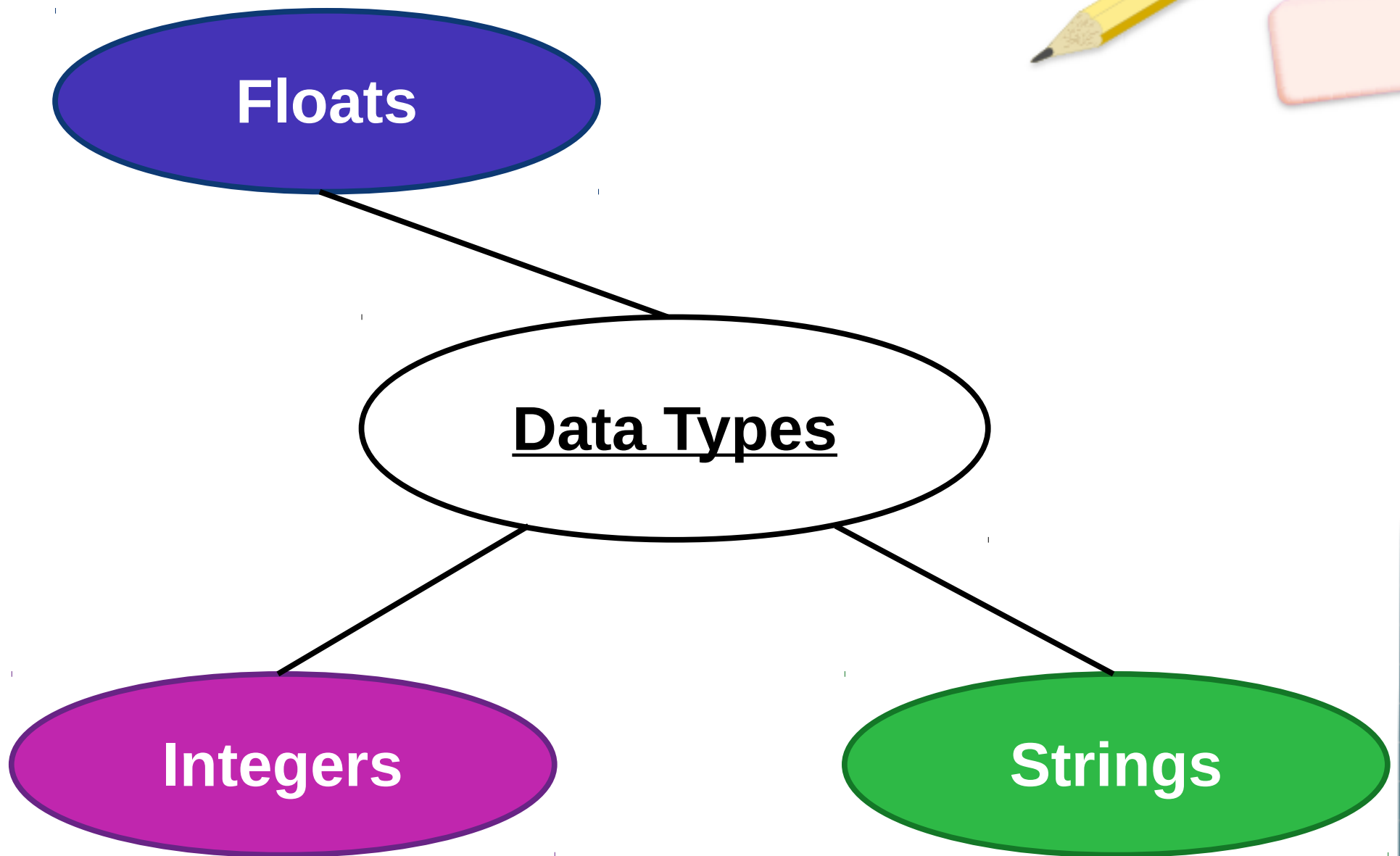
```
baba_lao = 6.02 * 10**23

part1 = baba_lao*5
part2 = baba_lao*23
part3 = baba_lao*303
part4 = baba_lao*4000
```




Funny  
...but hard to  
understand  
for different  
person

**Okay**



# Floats

Float: A number with a decimal



```
In [21]: a = 5.6
         type(a)
Out[21]: float
```

They work regularly with:

- Addition
- Subtraction
- Multiplication
- Division
- Exponents
- Anything mathematical


```
In [23]: a = 5.6
         add = a + 1
         subtract = a - 1
         mult = a * 2
         div = a / 2
         exponent = a**2
         complicated_math = (a**3 / 7*a) + (109 / a**(8))
         complicated_math
Out[23]: 140.4929126995759
```

# Integers

Integer: A number with no decimal

They work regularly with:

- Addition
- Subtraction
- Multiplication
- Division
- Exponents
- Anything mathematical



```
In [24]: a = 5
         type(a)

Out[24]: int
```

```
In [29]: a = 5
         add = a + 1
         subtract = a - 1
         mult = a * 2
         div = a / 2
         exponent = a**2

         complicated_math = (a**3) - (20 + (a * 10))
         complicated_math

Out[29]: 55
```



# Quick Question!

In [30]: ▶

```
a = 5.0  
type(a)
```

Out[30]:

Integer or Float?

# Strings



Surrounded by 'string' or "string"

```
In [31]: a = 'cat'
          type(a)

Out[31]: str
```

String: Something interpreted as text

They have interesting rules:

## You can:

- Add a string and a string
- Multiply a string and an integer

## You cannot:

- Add a float/integer to a string
- Multiply a string and a string/float
- Subtract
- Divide
- Exponents

# Strings

## Good

string + string

```
In [43]: a = 'cat'
         b = '3'
         a + b
Out[43]: 'cat3'
```

string \* integer

(integer \* string)  
is the same

```
In [44]: a = 'cat'
         b = 3
         a * b
Out[44]: 'catcatcat'
```

## Bad (error)

```
In [46]: a = 'cat'

a + 3      # Str + int
a + 3.0    # Str + float

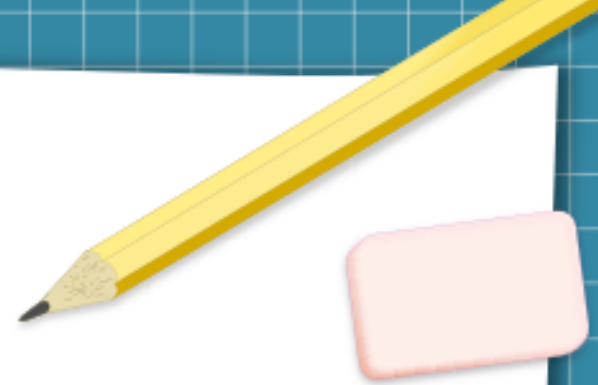
a * 3.0    # Str x float
a * '3'    # Str x Str

a - 3      # Str - int
a - 3.0    # Str - float
a - '3'    # Str - Str

a / 3      # Str / int
a / 3.0    # Str / float
a / '3'    # Str / Str

a**3       # Str ^ int
a**3.0     # Str ^ float
a**'3'     # Str ^ Str
```

# Strings



You can combine rules that work with strings!

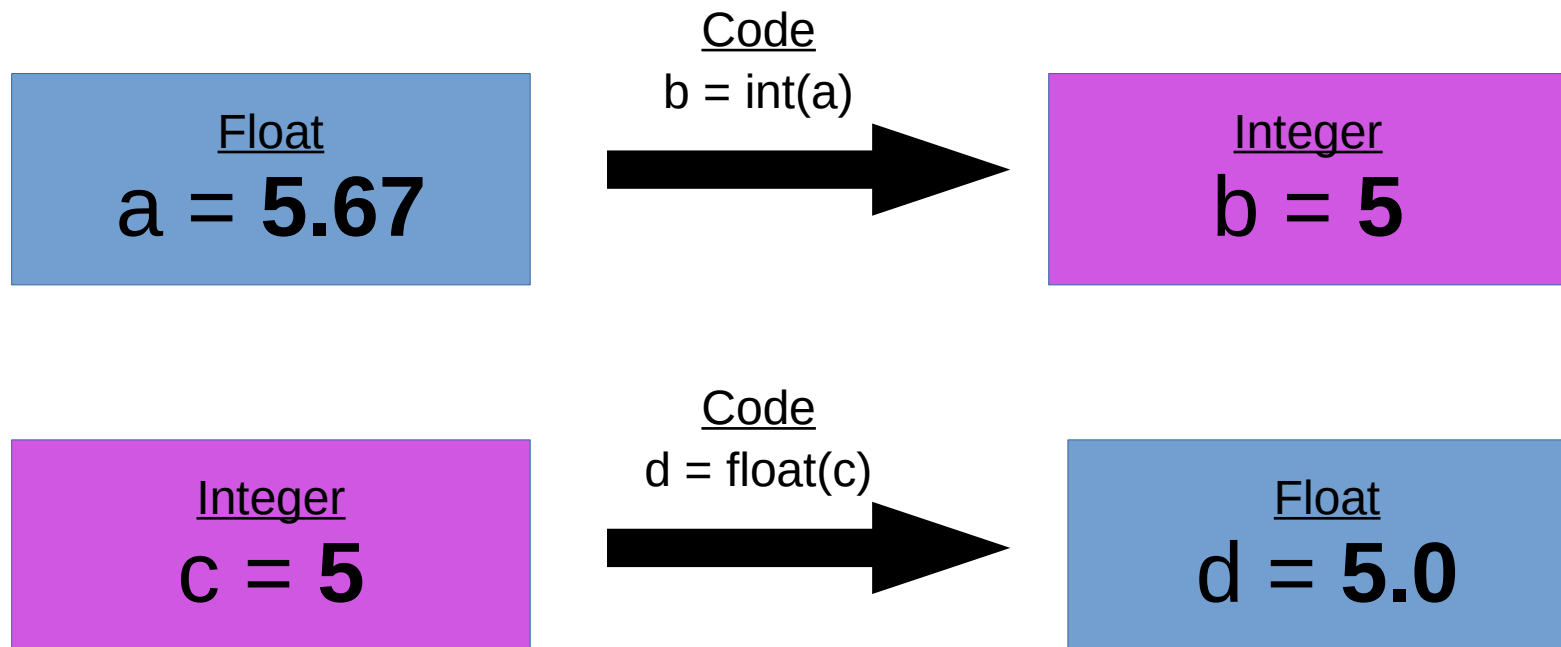
```
In [52]: ▶ a = 'cat'
          ( (a+' ') * 5) + 'end'
```

Out[52]: What am I?

```
In [60]: ▶ a = 'cat'
          b = ' '
          c = 5
          d = 'end'
          ((a+b)*c) + d
```

Out[60]: What am I?

# Converting: floats & integers



Practice:

`float( int(8.9999) ) = ?`

# Converting: strings

Float  
**a = 5.67**

string\_a = str(a)

String  
**string\_a = '5.67'**

Integer  
**b = 5**

string\_b = str(b)

String  
**string\_b = '5'**

String  
**c = '5.0'**

float\_c = float(c)

Float  
**float\_c = 5.0**

String  
**d = '5'**

integer\_d = int(d)

Integer  
**integer\_d = 5**

# User Input

## 1. Code:

```
input('What is your name? ')
```

## 2. After hitting enter, user can interact

```
In [*]: input('What is your name? ')
```

What is your name?

## 3. User interacts and hits enter

```
In [*]: input('What is your name? ')
```

What is your name?

PJ

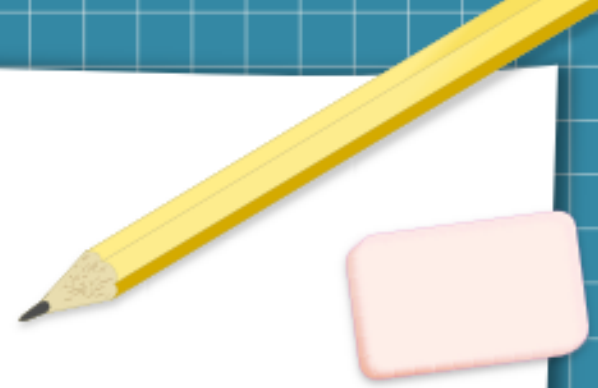
## 4. Data is shown

```
In [17]: input('What is your name? ')
```

What is your name? PJ

```
Out[17]: 'PJ'
```

# Saving User Input



User  
input

```
In [19]:  ▶ name = input('What is your name?  ')\n          What is your name?  PJ
```

Data  
stored

```
In [29]:  ▶ name\nOut[29]:  'PJ'
```

Note  
All inputs are strings

We can  
use it  
now!

```
In [27]:  ▶ 'his name is ' + name + ' and he is fresh'\nOut[27]:  'his name is PJ and he is fresh'
```



# Printing

Insert a string here

Use `print( )` to print something in the output line.

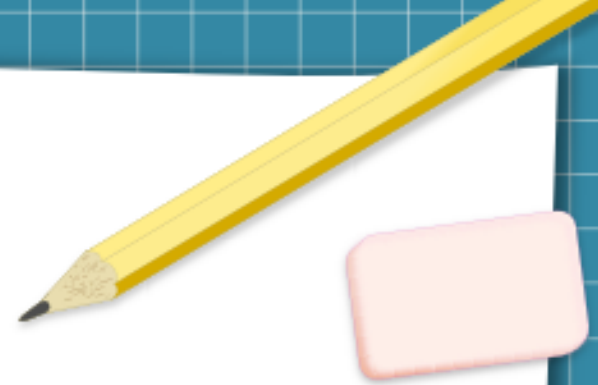
```
In [28]: ▶ print(name)  
PJ
```

'\n' represents a new line

```
In [34]: ▶ print(name+'\n'+is+'\n'+cool')  
PJ  
is  
cool
```

This is the best way to display outputs

# Printing vs. Output



## Printing

```
In [28]: ▶ print(name)  
PJ
```

```
In [34]: ▶ print(name+'\n'+is+'\n'+cool')  
PJ  
is  
cool
```

## Output

```
In [29]: ▶ name  
Out[29]: 'PJ'
```

```
In [35]: ▶ (name+'\n'+is+'\n'+cool')  
Out[35]: 'PJ\nis\ncool'
```

More things you can do → **Better**

# Challenge #1a

**Ask a user:**

- What is your name?
- Where are you from?
- What is your favorite subject to study?  
(any other questions)

**Your program should:**

- Write a paragraph about the user describing them

# Challenge #1b

Choose a Physics equation that you like (ex:  $F=ma$ )

## Ask a user:

- For the variables on the right side of the equation
  - Ex: What is the mass?
  - Ex: What is the acceleration?

## Your program should:

- State in a complete sentence the solution:
  - Ex: For a mass of \_\_\_\_ and an acceleration of \_\_\_\_, the force is \_\_\_\_.

# Challenge #1c

## Ask a user:

- Their name
- Their year of birth (ex: 1990)

## Your program should:

- Greet them by name
- Tell them the year that they will turn 100 years old

### Additional Bonus:

Ask the user to choose a number (n) between 1-10

Print their name (n) number of times

- Casino Royal – Stopped at 21:22

