Get ready:

On the whiteboard tables, please write the output of the following statements.
(**What will happen when you run these?)**

```
> print 30/2
> print ‘4+5*3’
> print ’Python is for snakes!’
> snake = ‘python’
  print (snake) + ‘is delicious’
> print (snakes)
```

Review:

Yesterday, you wrote your first program:

```
> print ‘Hello, world!’
  Hello, world!
```

Review:

Yesterday, you wrote your first program:

```
> print ‘Hello, world!’
  Hello, world!
```

Some of you experimented with other phrases:

```
> print ‘A group of crows is called a
murder.’
  A group of crows is called a murder.
```

Review:

Some of you also began to experiment with how you can use Python as a calculator.

```
> print 4+5
  9
> print 5 * 10
  50
> print 10/2
  5
> print 80-3993219459
  -3993219379
```

Today's lesson:

One of the most powerful things we can do in programming is create **variables**.

Variables hold a value.

Today's lesson:

One of the most powerful things we can do in programming is create **variables**.

Variables hold a value.

Variables hold a value.

Variables hold a value.

Variables hold a value.

Variables hold a value.

Today’s lesson:

For example,

```
> name = ‘Jenkins’
  print (name)
```

```
 Jenkins
```

Today’s lesson:

Variables can store different types of information:

```
> phone_number = ‘5109227781’
  print (phone_number)

 5109227781
```

Today’s lesson:

Variables can store different types of information:

```
> phone_number = ‘5109227781’
  print (phone_number)

 5109227781
```

Tomorrow, we’ll talk about the difference between this phone number and the number 5,109,227,781

Today's lesson:

But why are variables so great, anyway?

If we know someone's name is **Jenkins**, then why do we even need a variable called **name**? Why don't we just use **Jenkins**?

Today's lesson:

Because sometimes, we need our variables to change.

A shirt costs $20 plus 8.5% tax.
Pants cost $25 plus 8.5% tax.
A hat costs $18 plus 8.5% tax.

```
> print  20 + 20*(8.5/100)
> print 25 + 25*(8.5/100)
> print 18 + 18*(8.5/100)
```

Today’s lesson:

```
tax = 8.5/100
```

A shirt costs $20 plus 8.5% tax.
Pants cost $25 plus 8.5% tax.
A hat costs $18 plus 8.5% tax.

```
> print  20 + 20*(tax)
> print 25 + 25*(tax)
> print 18 + 18*(tax)
```

Today's lesson:

What if tax goes up to 9.5%?

```
tax = 8.5/100
```

A shirt costs $20 plus 9.5% tax.
Pants cost $25 plus 9.5% tax.
A hat costs $18 plus 9.5% tax.

```
> print  20 + 20*(tax)
> print 25 + 25*(tax)
> print 18 + 18*(tax)
```

Today's lesson:

Just change one number!

```
tax = 9.5/100
```

A shirt costs $20 plus 9.5% tax.
Pants cost $25 plus 9.5% tax.
A hat costs $18 plus 9.5% tax.

```
> print  20 + 20*(tax)
> print 25 + 25*(tax)
> print 18 + 18*(tax)
```

Today's lesson:

Just change one number!

```
tax = 9.5/100
```

A shirt costs $20 plus 9.5% tax.
Pants cost $25 plus 9.5% tax.
A hat costs $18 plus 9.5% tax.

```
> print  20 + 20*(tax)
> print 25 + 25*(tax)
> print 18 + 18*(tax)
```

Expression stays the same!

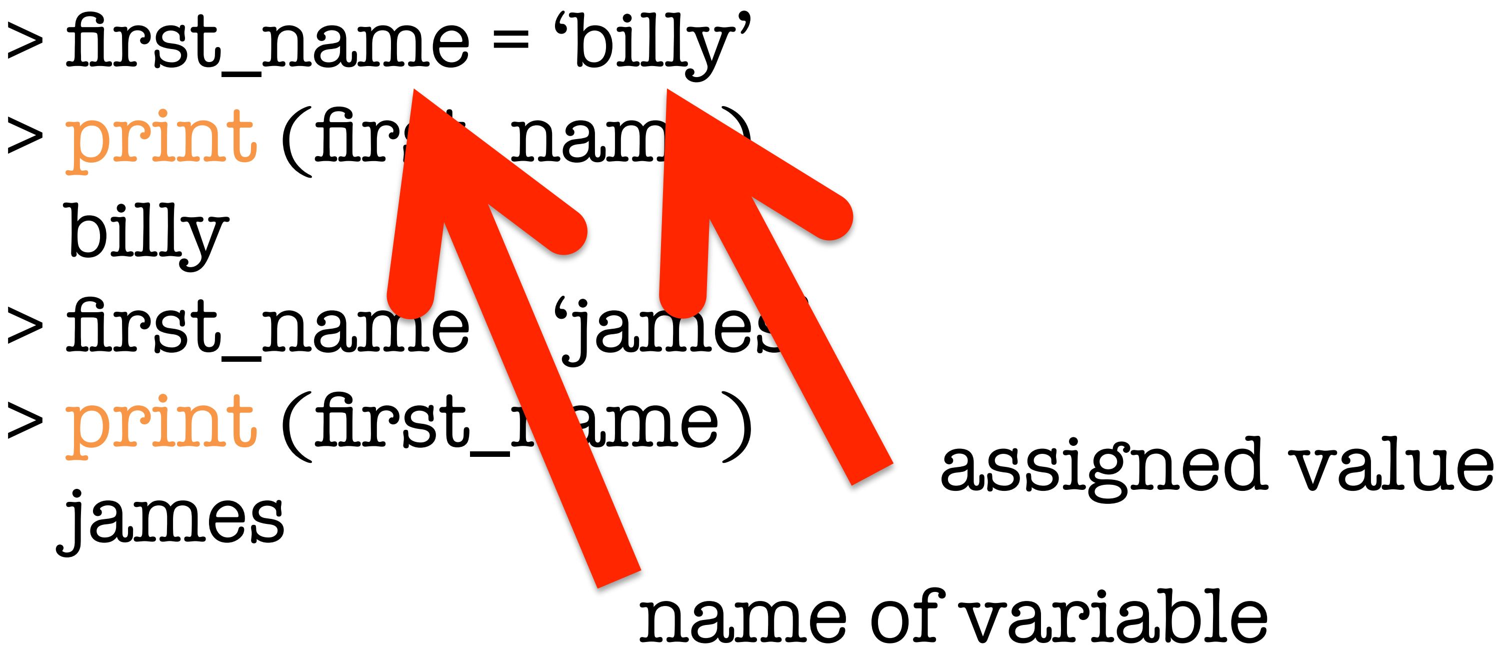
Today’s lesson:

The values of variables can change **even if their name stays the same.**

```
> first_name = ‘billy’
> print (first_name)
  billy
> first_name = ‘james’
> print (first_name)
  james
```

Today’s lesson:

The values of variables can change **even if their name stays the same.**



```
> first_name = ‘billy’
> print (first_name)
  billy
> first_name = ‘james’
> print (first_name)
  james
```

assigned value

name of variable

Today’s lesson:

Finally, sometimes it’s useful for a user to give you a value for a variable. For example:

```
> age = raw_input(‘How old are you?’)
> print age
  16
```

raw_input is a very useful function because it allows you to collect information from a user.

Your turn:

Please visit our website at www.REALM-CS.com to find the exercises for today.

Focus on exercises **3a** (Variable challenges).
Do as many as you can.
Make mistakes. Break it.
Try: "What happens if I …."