NUMBERS AS VARIABLES

Variables can also be numbers. When assigning a number value to a variable, DO NOT surround the number with quotation marks.

```
number = 5
string = "five"
string = "5"
```

A number can be a string if you surround it with quotation marks. But you can't do math with a string.

CALCULATIONS WITH NUMBER VARIABLES

In Python, you can use numbers to calculate math problems. For example, you can create a homework helper program where Python does the calculations.

You can make two number variables, add them together, and then print the sum like this:

Remember to assign the values of "num1" and "num2" *before* you use them in a calculation for "sum." You can't use variables until *after* you assign them values.

```
num1 = 4
num2 = 5
sum = num1 + num2
print(sum)
```

This will print: 9

NUMBER TYPES

Python uses different types of numbers. Two of those types are **INTEGERS** and **FLOATS**.

Integers

In Python, **integers** are positive or negative whole numbers that don't use a decimal place.

You can create a variable called "num1" and assign it an integer value of 3:

```
num1 = 3
```
A whole number

String variables, even if they have whole numbers inside them, aren't recognized by Python as integers.

FOR EXAMPLE, when you add two strings together, they are combined as one string:

```
num1 = "4"
num2 = "3"
sum = num1 + num2
print(sum)
```

Quotation marks around the number means it's a string.

This will print 43, which is the string 4 and the string 3 combined into a single string.

Sometimes it's necessary to convert a variable to an **integer data type**.

FOR EXAMPLE, if you ask a user to enter a number using the **input()** function, Python stores their answer as a string.

```
width = input("Enter the rectangle's width. ")
height = input("Enter the rectangle's height. ")
```

In the program above, if the user entered 5 and 8, the numbers would be stored as strings—"5" and "8"—and not as integers.

Convert the user's answers from strings to integers using the int() function. You can then perform calculations with them.

Multiply the length and width of a rectangle to find the area like this:

> Width and height start out as strings.

```
width = input("Enter the rectangle's width. ")
height = input("Enter the rectangle's height. ")
width = int(width)
height = int(height)
print(width*height)
```

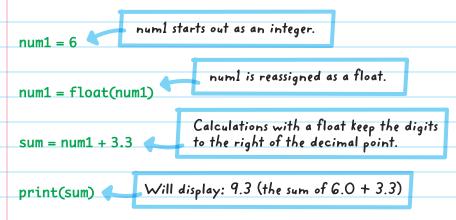> Width and height are reassigned as integers.

> The number values for width and height can now be used in calculations, such as finding the area.

Floats

Floats (or floating point numbers) are numbers that have a decimal point.

FOR EXAMPLE, the values to these variables are floats:

```
num1 = 3.3
num2 = 6.0
num3 = 2.9564576
```

> FLOATS

To convert a variable value from an integer or string to a float, use the **float( )** function like this:

```
num1 = 6
```
num1 starts out as an integer.

```
num1 = float(num1)
```
num1 is reassigned as a float.

```
sum = num1 + 3.3
```
Calculations with a float keep the digits to the right of the decimal point.

```
print(sum)
```
Will display: 9.3 (the sum of 6.0 + 3.3)

NUMBER CONVERSION

int( ): Converts a value to an integer. Useful for converting a user's input string into an integer to do math.

float( ): Converts a value to a floating point number (number with a decimal point). Useful when you are using values that are not whole numbers.

MATHEMATICAL EXPRESSIONS

Numbers and math are used in programming to represent the location of a character in a game, calculate dates, set timers, display colors, calculate scores, and more. **VALUES** and **OPERATORS** combine to make **EXPRESSIONS**.

EXPRESSION

A combination of values and operators that can be evaluated to a single value. For example, $2 + 3$ is an expression that evaluates to (or equals) 5.

OPERATOR
$2 + 3$
VALUES

EXAMPLES OF EXPRESSIONS:

$4 * 3$

$3 / 4 * 5 - 5 / 2$

$(7 * 3) - (5 - 7) / (4 + 5)$

Variables can also hold number values and be used in mathematical expressions. For example, if a variable called "x" holds the value of 4, it can be used in an expression:

VALUE

A piece of information. In mathematical expressions, values are numbers.

$x + 6$ Because x holds the value 4, this expression evaluates to 10.

There are seven common math operators:

NAME	SYMBOL	EXAMPLE
addition	+	5 + 2
subtraction	−	5 – 2
multiplication	*	5 * 2
division	/	5 / 2
modulus	%	5 % 2
exponent	**	5 ** 2
floor division	//	5 // 2

The remainder portion of a quotient. For example, 5 ÷ 2 = 2 with a remainder, or **modulus**, of 1.

Divides two numbers and ignores the remainder. For example, 5 ÷ 2 = 2 (the remainder, 1, is ignored).

The symbols "x" and "÷" are not used in Python.

ORDER OF OPERATIONS

Use the **ORDER OF OPERATIONS** when solving equations:

ORDER OF OPERATIONS
A set of rules for deciding the order that operations will be evaluated

1. Solve any parts of the equation that are inside brackets or parentheses. If there is more than one set of parentheses, solve the innermost equation first.

Parentheses can be *nested*. This means that there can be one set of parentheses inside another set, like this: $((2 + 18) * 5) / 10$.

The innermost equation is $2 + 18$ and should be solved first.

2. Calculate exponents, absolute value, and square roots from left to right.

3. Solve all the multiplication and division from left to right.

4. Complete the addition and subtraction from left to right.

To remember the order of operations, think "**P**lease **E**xcuse **M**y **D**ear **A**unt **S**ally," or **PEMDAS** (**P**arentheses, **E**xponents, **M**ultiplication, **D**ivision, **A**ddition, **S**ubtraction). But be careful, because PEMDAS isn't foolproof. For example, you should do subtraction before addition if you're calculating from left to right—and the same goes for division and multiplication.

First, solve the equation in parentheses: 2 * (6 – 3)

           6 – 3 = 3   2 * (3)

Then, multiply to get your answer: 2 * 3 = 6

If you have the same expression
without the parentheses . . .   2 * 6 – 3

First, multiply: 2 * 6 = 12   12 – 3

Then, subtract to get your answer: 12 – 3 = 9

Even though they contain the same numbers, the two equations have different answers because of the order of operations. Python will always follow the order of operations when it returns a value.

Printing Mathematical Expressions

If an expression is inside the print function, then Python will print the value of the expression to the shell window.

FOR EXAMPLE:

`print(5 * 3)`   This will print: 15

1. In Python, what does the ** operator do?

2. To find the remainder of a division, you would use the
_____ operator.

3. What will each expression return?
   A. 5 // 4
   B. 3 * (5 + 2) – 6
   C. 2 * 5
   D. 4 / 2
   E. 6 % 2
   F. 6 / 3 + (2 – 1) + 2 * 3

4. Write out the code to print the answer to 5 times 2.

5. In the correct order of operations, which comes first:
exponents or addition?

6. What is an expression?

7. In mathematical expressions, values are always (numbers / symbols).

8. If you want to find the whole-number value of a division operation **without** the remainder, you should use the _____ operator.

9. What will the following code print?

```
width = 4
height = 12
perimeter = width * 2 + height * 2
print(perimeter)
```

10. Where is the error in this program?

```
dog_years = human_years * 7
human_years = 3
print(dog_years)
```

11. What's the difference between an integer and a float variable?

12. Will the following program print an integer or a float number?

```
d = 4.2
t = 3
speed = d / t
print(speed)
```

13. What function should you use to convert a variable to an integer?

CHECK YOUR ANSWERS

1. ** is used as the exponent operator.

2. Modulus or %

3. **A.** 1
B. 15
C. 10
D. 2
E. 0
F. 9

4. print(5 * 2)

5. Exponents

6. An expression is the combination of values and operators that can be evaluated to a single value.

7. Numbers

8. Floor or //

9. 32

10. The problem is "human_years" is assigned the value of 3 after it is used in the first line (human_years * 7).

11. An integer is a whole number that does not have a decimal, and a float is a number that has a decimal.

12. Float, because "d" is a float and used to calculate "speed"

13. int( )