STATEMENTS AND EXPRESSIONS

An Introduction to Computer Science

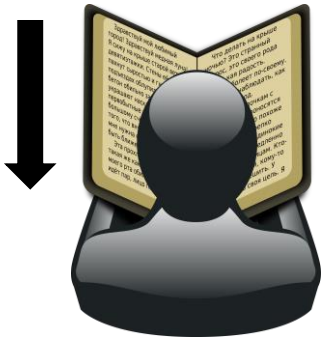
Let's learn about Statements and Expressions.

Sequence

1st	`count = 0`
2nd	`print(count)`
3rd	`count = count + 5`
4th	`count = count + 2`
5th	`print(count)`

Programs execute line by line, one step at a time.
Many times, the programs run super fast, and you won't see this behavior.
We have some tools that let us slow down and see the execution.
But either way, programs run one line at a time.

Metaphor



Think about reading a book or acting out a script, how we move down a page.
This is just like how the instructions of a program are read.

Time

```
count = 0
print(count)
count = count + 5
count = count + 2
print(count)
```

```
Line 1 took 1 millisecond
Line 2 took 3 milliseconds
Line 3 took 2 milliseconds
Line 4 took 2 millliseconds
Line 5 took 3 millliseconds
Total execution time: 11 milliseconds
```

The result of this step-by-step instruction is that programs are executed *over time*.
Although execution happens very, very fast, it does not happen all at once.
Time is a crucial component in running programs.

Statements

- Assignment Statement

```
count = 0
```

- Print statement

```
print(count)
```

A single line of code is known as a "Statement".
A program is, quite literally, a sequence of statements.
There are many kinds of statements, but we only know about a few so far.

Expressions Can Have

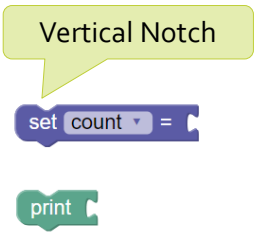
- Literal Values
  `5, 10.0, True, "Hello", None`
- Arithmetic operators
  `1 + 2, 4 - 5, 8 / 4`
- Boolean operators
  `5 > 4, False or not True`
- Variables
  `age, score, my_string_variable`
- And more expressions
  - `(1 + 5) > age and not True`

Expressions inside of expressions inside of expressions!

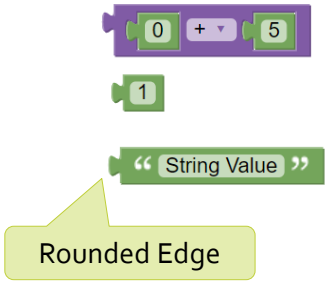
An expression is any kind of combination of literal values, arithmetic operators, Boolean operators, and variables.
Any value is an expression.
Any variable is an expression.
Expressions can be made up of expressions.

Connecting Statements and Expressions

Statements



Expressions



In BlockPy, you can see that expression blocks have a rounded left edge. Statement blocks have a pointy bit on the bottom and a notch on the top. This visually shows you the difference between statements and expressions. Statements join vertically, and expressions join horizontally.

Order is Important

Chapter 15

Chapter 1

Prologue

Chapter 5

Since programs run from top to bottom, the order of statements is important.
If you rearranged the sentences of a book at random, they wouldn't make much sense - the same is true for programs.

Evaluation

The Code

```
print((5 + 3) * 4 > 8)
```

In the Computer's Head

- First, `5+3` is replaced with `8`
- Next, `8*4` is replaced with `32`
- Then, `32>8` is replaced with `True`
- Lastly, `True` is printed to the console

We say that the computer "Evaluates" an expression, when it reduces it.
The "evaluation" doesn't happen until the program is run.
It is important to learn how to "evaluate" expressions in your head.

Order within Expressions

Math

```
5 * (4 + 2)
```

First, 4+2 is evaluated to 6
Then, 5*6 is evaluated to 30

Assignment

```
score = 5 + 3
```

First, 5+3 is evaluated to 8
Then, 5 is stored in `score`

Following the order within expressions is tricky, because they are not always left to right. Consider this mathematical expression, where the addition happens before the multiplication.
The same thing happens with an assignment; the expression on the right is evaluated BEFORE the assignment occurs on the left.