# Short Course on Programming

## 2. Fundamental Programming Principles I: Variables, Data Types & Logic

Ronni Grapenthin

**August 28, 2017**

YOU'LL NEVER FIND A PROGRAMMING LANGUAGE THAT FREES YOU FROM THE BURDEN OF CLARIFYING YOUR IDEAS.
BUT I KNOW WHAT I MEAN!

"The Uncomfortable Truths Well",
http://xkcd.com/568 (April 13, 2009)

## How does (computer) programming work?

Well, fist we should clearify terminology here!

What is a programming language?

What is a program?

## Alright, what is it then?

### Definitions (broad sense)

A **programming language** is an unambiguous artificial language that is made up of a set of symbols (vocabulary) and grammatical rules (syntax) to instruct a machine.

A **program** is a set of instructions in one or multiple programming languages that specifies the behavior of a machine.

**Compilation** or **interpretation** is the verification of a program and its translation into machine readable instructions of a specific platform.

## Programming languages . . . continued

Two broad families can be identified:

**Interpreted languages**
An interpreter program is necessary to take in commands, check syntax and translate to machine language at runtime (e.g., Matlab, Unix Shell)

## Programming languages . . . continued

Two broad families can be identified:

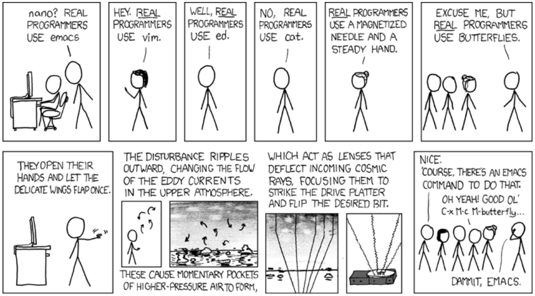
**Interpreted languages**
An interpreter program is necessary to take in commands, check syntax and translate to machine language at runtime (e.g., Matlab, Unix Shell)

**Compiled languages**
Programs are translated and saved in machine language. At runtime no additional program is necessary (e.g., C/C++).

## Now, how does programming work?

## Now, how does programming work?

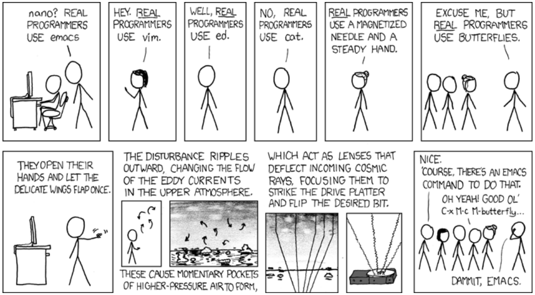
open **text** editor (vi, notepad, . . . , not MS Word!)


nano? REAL PROGRAMMERS USE emacs
HEY. REAL PROGRAMMERS USE vim.
WELL, REAL PROGRAMMERS USE ed.
NO, REAL PROGRAMMERS USE cat.
REAL PROGRAMMERS USE A MAGNETIZED NEEDLE AND A STEADY HAND.
EXCUSE ME, BUT REAL PROGRAMMERS USE BUTTERFLIES.
THEY OPEN THEIR HANDS AND LET THE DELICATE WINGS FLAP ONCE.
THE DISTURBANCE RIPPLES OUTWARD, CHANGING THE FLOW OF THE EDDY CURRENTS IN THE UPPER ATMOSPHERE.
THESE CAUSE MOMENTARY POCKETS OF HIGHER-PRESSURE AIR TO FORM,
WHICH ACT AS LENSES THAT DEFLECT INCOMING COSMIC RAYS, FOCUSING THEM TO STRIKE THE DRIVE PLATTER AND FLIP THE DESIRED BIT.
NICE.
'COURSE, THERE'S AN EMACS COMMAND TO DO THAT.
OH YEAH! GOOD OL' C-x M-c M-butterfly...
DAMMIT, EMACS.

http://www.xkcd.com/378/

## Now, how does programming work?

open **text** editor (vi, notepad, . . . , not MS Word!)

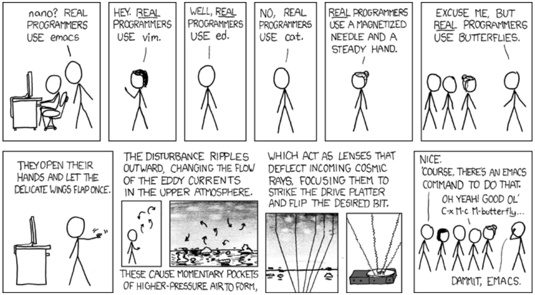

nano? REAL PROGRAMMERS USE emacs
HEY. REAL PROGRAMMERS USE vim.
WELL, REAL PROGRAMMERS USE ed.
NO, REAL PROGRAMMERS USE cat.
REAL PROGRAMMERS USE A MAGNETIZED NEEDLE AND A STEADY HAND.
EXCUSE ME, BUT REAL PROGRAMMERS USE BUTTERFLIES.
THEY OPEN THEIR HANDS AND LET THE DELICATE WINGS FLAP ONCE.
THE DISTURBANCE RIPPLES OUTWARD, CHANGING THE FLOW OF THE EDDY CURRENTS IN THE UPPER ATMOSPHERE.
THESE CAUSE MOMENTARY POCKETS OF HIGHER-PRESSURE AIR TO FORM,
WHICH ACT AS LENSES THAT DEFLECT INCOMING COSMIC RAYS, FOCUSING THEM TO STRIKE THE DRIVE PLATTER AND FLIP THE DESIRED BIT.
NICE.
'COURSE, THERE'S AN EMACS COMMAND TO DO THAT.
OH YEAH! GOOD OL' C-x M-c M-butterfly...
DAMMIT, EMACS.

http://www.xkcd.com/378/

translate your (mental) flowchart into a set of instructions according to the rules of an applicable programming language

## Now, how does programming work?

open **text** editor (vi, notepad, . . . , not MS Word!)


nano? REAL PROGRAMMERS USE emacs
HEY. REAL PROGRAMMERS USE vim.
WELL, REAL PROGRAMMERS USE ed.
NO, REAL PROGRAMMERS USE cat.
REAL PROGRAMMERS USE A MAGNETIZED NEEDLE AND A STEADY HAND.
EXCUSE ME, BUT REAL PROGRAMMERS USE BUTTERFLIES.
THEY OPEN THEIR HANDS AND LET THE DELICATE WINGS FLAP ONCE.
THE DISTURBANCE RIPPLES OUTWARD, CHANGING THE FLOW OF THE EDDY CURRENTS IN THE UPPER ATMOSPHERE.
THESE CAUSE MOMENTARY POCKETS OF HIGHER-PRESSURE AIR TO FORM,
WHICH ACT AS LENSES THAT DEFLECT INCOMING COSMIC RAYS, FOCUSING THEM TO STRIKE THE DRIVE PLATTER AND FLIP THE DESIRED BIT.
NICE.
'COURSE, THERE'S AN EMACS COMMAND TO DO THAT.
OH YEAH! GOOD OL' C-x M-c M-butterfly...
DAMMIT, EMACS.

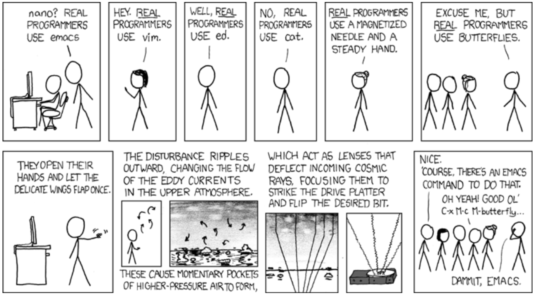
http://www.xkcd.com/378/

translate your (mental) flowchart into a set of instructions according to the rules of an applicable programming language

test your program for syntactical correctness (ask interpreter/compiler)

## Now, how does programming work?

open **text** editor (vi, notepad, . . . , not MS Word!)


nano? REAL PROGRAMMERS USE emacs
HEY. REAL PROGRAMMERS USE vim.
WELL, REAL PROGRAMMERS USE ed.
NO, REAL PROGRAMMERS USE cat.
REAL PROGRAMMERS USE A MAGNETIZED NEEDLE AND A STEADY HAND.
EXCUSE ME, BUT REAL PROGRAMMERS USE BUTTERFLIES.
THEY OPEN THEIR HANDS AND LET THE DELICATE WINGS FLAP ONCE.
THE DISTURBANCE RIPPLES OUTWARD, CHANGING THE FLOW OF THE EDDY CURRENTS IN THE UPPER ATMOSPHERE.
THESE CAUSE MOMENTARY POCKETS OF HIGHER-PRESSURE AIR TO FORM,
WHICH ACT AS LENSES THAT DEFLECT INCOMING COSMIC RAYS, FOCUSING THEM TO STRIKE THE DRIVE PLATTER AND FLIP THE DESIRED BIT.
NICE.
'COURSE, THERE'S AN EMACS COMMAND TO DO THAT.
OH YEAH! GOOD OL' C-x M-c M-butterfly...
DAMMIT, EMACS.

http://www.xkcd.com/378/

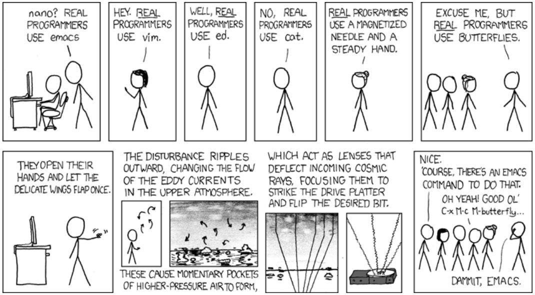
translate your (mental) flowchart into a set of instructions according to the rules of an applicable programming language

test your program for syntactical correctness (ask interpreter/compiler)

if errors, fix them and go back to (3)

## Now, how does programming work?

open **text** editor (vi, notepad, . . . , not MS Word!)


nano? REAL PROGRAMMERS USE emacs
HEY. REAL PROGRAMMERS USE vim.
WELL, REAL PROGRAMMERS USE ed.
NO, REAL PROGRAMMERS USE cat.
REAL PROGRAMMERS USE A MAGNETIZED NEEDLE AND A STEADY HAND.
EXCUSE ME, BUT REAL PROGRAMMERS USE BUTTERFLIES.
THEY OPEN THEIR HANDS AND LET THE DELICATE WINGS FLAP ONCE.
THE DISTURBANCE RIPPLES OUTWARD, CHANGING THE FLOW OF THE EDDY CURRENTS IN THE UPPER ATMOSPHERE.
THESE CAUSE MOMENTARY POCKETS OF HIGHER-PRESSURE AIR TO FORM,
WHICH ACT AS LENSES THAT DEFLECT INCOMING COSMIC RAYS, FOCUSING THEM TO STRIKE THE DRIVE PLATTER AND FLIP THE DESIRED BIT.
NICE.
'COURSE, THERE'S AN EMACS COMMAND TO DO THAT.
OH YEAH! GOOD OL' C-x M-c M-butterfly...
DAMMIT, EMACS.

http://www.xkcd.com/378/

translate your (mental) flowchart into a set of instructions according to the rules of an applicable programming language

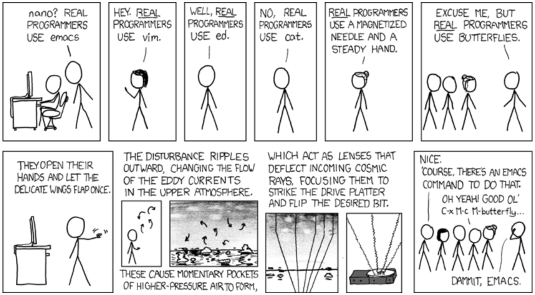
test your program for syntactical correctness (ask interpreter/compiler)

if errors, fix them and go back to (3)

test your program for semantical errors (the fun part!)

## Now, how does programming work?

open **text** editor (vi, notepad, . . . , not MS Word!)


nano? REAL PROGRAMMERS USE emacs
HEY. REAL PROGRAMMERS USE vim.
WELL, REAL PROGRAMMERS USE ed.
NO, REAL PROGRAMMERS USE cat.
REAL PROGRAMMERS USE A MAGNETIZED NEEDLE AND A STEADY HAND.
EXCUSE ME, BUT REAL PROGRAMMERS USE BUTTERFLIES.
THEY OPEN THEIR HANDS AND LET THE DELICATE WINGS FLAP ONCE.
THE DISTURBANCE RIPPLES OUTWARD, CHANGING THE FLOW OF THE EDDY CURRENTS IN THE UPPER ATMOSPHERE.
THESE CAUSE MOMENTARY POCKETS OF HIGHER-PRESSURE AIR TO FORM,
WHICH ACT AS LENSES THAT DEFLECT INCOMING COSMIC RAYS, FOCUSING THEM TO STRIKE THE DRIVE PLATTER AND FLIP THE DESIRED BIT.
NICE.
'COURSE, THERE'S AN EMACS COMMAND TO DO THAT.
OH YEAH! GOOD OL' C-x M-c M-butterfly...
DAMMIT, EMACS.

http://www.xkcd.com/378/

translate your (mental) flowchart into a set of instructions according to the rules of an applicable programming language

test your program for syntactical correctness (ask interpreter/compiler)

if errors, fix them and go back to (3)

test your program for semantical errors (the fun part!)

if errors, fix them and go back to (3)

## Variables (1)

### Definitions – a selection

**Donald Knuth**: A quantity that may possess different values as a program is being executed.
**Mehran Sahami**: A box in which we stuff things – i.e. a box with variable content.
**Wikipedia**: User defined keyword that is linked to a value stored in computer's **memory** (runtime).

The concept of a **variable** consists of:

## Variables (1)

**Definitions – a selection**

**Donald Knuth**: A quantity that may possess different values as a program is being executed.
**Mehran Sahami**: A box in which we stuff things – i.e. a box with variable content.
**Wikipedia**: User defined keyword that is linked to a value stored in computer's **memory** (runtime).

The concept of a **variable** consists of:

- name
- type
- value

## Don’t even think that’s as simple as it sounds . . .

### ‘Hello World’ in Python

```
>>> prnt
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NameError: name 'prnt' is not defined


>>> print

>>> print "Hello
  File "<stdin>", line 1
    print "Hello
               ^
SyntaxError: EOL while scanning string literal


>>> print "Hello Wrld"
Hello Wrld


>>> print "Hello World"
Hello World
```

## Don’t even think that’s as simple as it sounds . . .

### ‘Hello World’ in Python

```
>>> prnt
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NameError: name 'prnt' is not defined
```

```
>>> print
```

```
>>> print "Hello
  File "<stdin>", line 1
    print "Hello
               ^
SyntaxError: EOL while scanning string literal
```

```
>>> print "Hello Wrld"
Hello Wrld
```

```
>>> print "Hello World"
Hello World
```

### ‘Hello World’ on Shell

```
[glacier:~] grapenthin% ech
ech: Command not found.
```

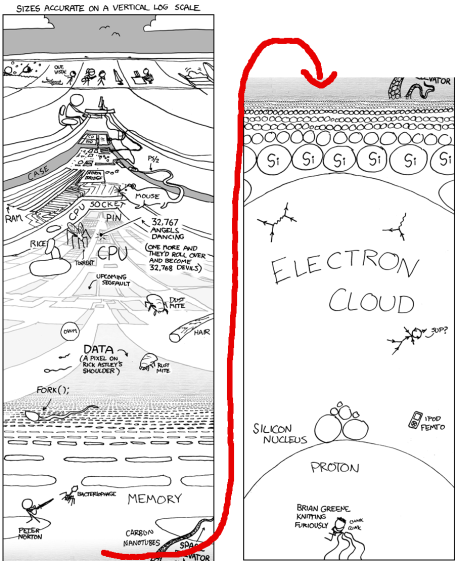
```
[glacier:~] grapenthin% echo
```

```
[glacier:~] grapenthin% echo "Hello
Unmatched ".
```

```
[glacier:~] grapenthin% echo "Hello Wrld"
Hello Wrld
```

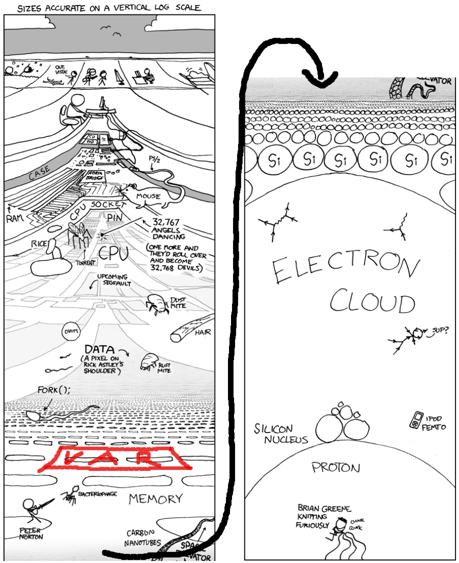
```
[glacier:~] grapenthin% echo "Hello World"
Hello World
```

## Memory interlude



“Depth”,
http://xkcd.com/485 (September 16, 2009)

# Memory interlude



"Depth",
http://xkcd.com/485 (September 16, 2009)

## Variables (2) – name

USE VALID NAME: follow programming language rules – Python variable names must **begin with a letter**, followed by any **combination of letters, digits, and underscores**. Uppercase different from lowercase. Don't use reserved keywords!

USE MEANINGFUL NAMES, i.e. names that speak:

'`lengthGlacier`' or '`glacier_length`' ... **Don't use** '`a`' – avoid ambiguity (Unless following a paper, textbook)

USE CONSITENT FORMATTING, i.e.: '`my_cool_var`' vs. '`myCoolVar`' –supports reading

Many style guides exist – punchline: use meaningful names, be consistent (that's hard enough)!

## Variables (3) – type

**What is a type?** – Think of sets of numbers in math: $\mathbb{N}, \mathbb{R}, \mathbb{Z}, \ldots$ The type refers to how **values** are being represented in a computer's memory, i.e. the meaning of each bit, and how many bits are necessary

### Two kinds of Types

- primitive, built-in types – for Python e.g.: `'boolean'`, `'int'`, `'float'`, `'complex'` (important for `print` function)
- non-primitive (built-in or self made) – sequences, iterators, classes, ... `https://docs.python.org/2.7/library/stdtypes.html`

### Types in Programming Languages

- some languages, e.g. Python, Shells, Matlab are weakly typed: implicit type conversions (OR one type can be treated as another)
- this is nice at first, occasionally this leads to nasty/hard to fix problems (e.g. string interpreted as number, etc.)

## Variables (4) – value

### Value

- a value of the type of the variable: `23, 3.1415926..., false`
- i.e., the thing we stuff in the box
- can/should change during the runtime of the program, otherwise use a constant (if possible)

### Declaring a variable and Assigning a value:

**In General:** `(type) name = value;` or `(type) name = expression;`

**Python:** `myNewVar=10;` **TC-Shell** (differs) `set myNewVar = 10;`
Access to the values (de-referencing):
**Python:** use `myNewVar;` **TC-Shell** (differs) use '`$`': `$myNewVar`

### What's that?

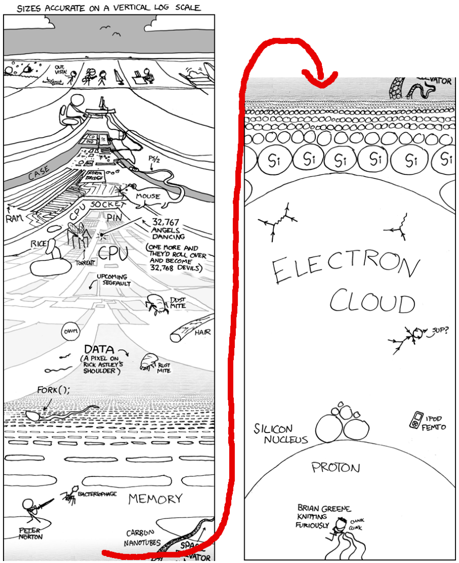
```
myNewVar = myNewVar + 1;
```

## Advanced Variables: Vectors and Matrices (1)

### Array variables

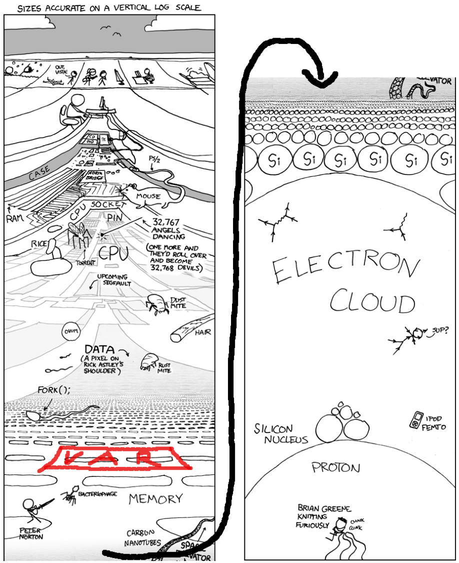
- are lists, vectors, matrices of data (1 to n dimensional – book keeping can become a hassle)
- therefore instead of one value they hold a **list of values**
- linked to a chunk of memory (a sequence of boxes)
- access by index number
- Difference between Python List and Numpy array!
- Shells allow only vectors.

# Memory interlude (2)



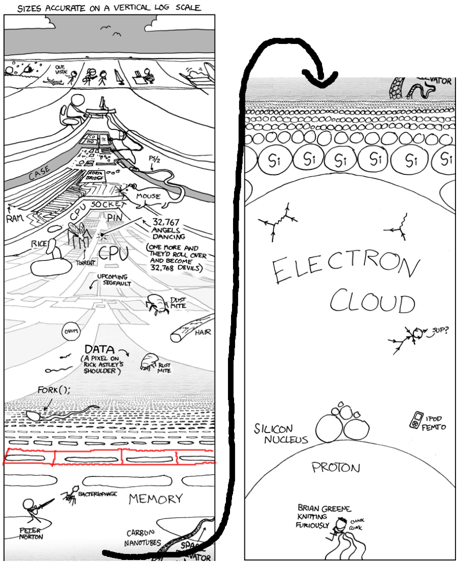
"Depth",
http://xkcd.com/485 (September 16, 2009)

## Memory interlude (2)



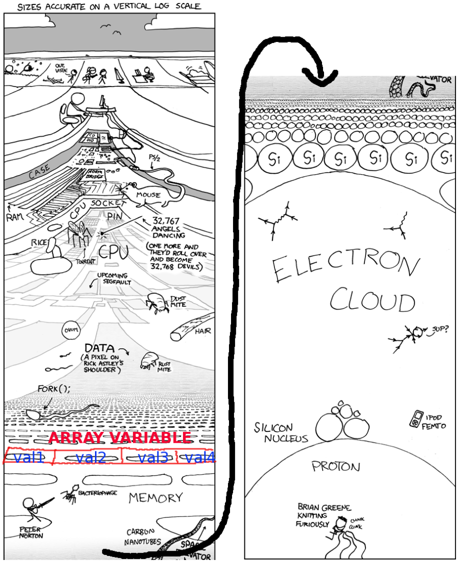
"Depth",
http://xkcd.com/485 (September 16, 2009)

## Memory interlude (2)



"Depth",
http://xkcd.com/485 (September 16, 2009)

# Memory interlude (2)



“Depth”,
http://xkcd.com/485 (September 16, 2009)

## Advanced Variables: Vectors and Matrices (2)

### Example: Numeric Vector

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| index: | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
| vector: | 12 | 23.3 | 23.3 | nan | nan | 1 | 42 | 42.1 | 23 | 5 | nan | nan | 0 | 0 | 0 |

## Advanced Variables: Vectors and Matrices (2)

### Example: Numeric List

| index: | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| vector: | 12 | 23.3 | 23.3 | nan | nan | 1 | 42 | 42.1 | 23 | 5 | nan | nan | 0 | 0 | 0 |

### Example: String

| index: | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| sting: | h | e | l | l | o |  | w | o | r | l | d | ! | ! | ! | ! |

```
>>> x="hello world!!!!"
>>> x(1)
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: 'str' object is not callable
>>> x[1]
'e'
```

## Logic 101

Use logic to connect multiple conditions and test for certain cases (0 is false, 1 is true):

## Logic 101

Use logic to connect multiple conditions and test for certain cases (0 is false, 1 is true):

**‘NOT’ (‘~’, ’!’):**

| a | !a |
|---|---|
| 0 | 1 |
| 1 | 0 |

## Logic 101

Use logic to connect multiple conditions and test for certain cases (0 is false, 1 is true):

**‘NOT’ (‘~’, ’!’):**

| a | !a |
|---|---|
| 0 | 1 |
| 1 | 0 |

**‘AND’ (‘&&’):**

| a | b | a && b |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

## Logic 101

Use logic to connect multiple conditions and test for certain cases (0 is false, 1 is true):

**'NOT' ('~', '!'):**

| a | !a |
|---|---|
| 0 | 1 |
| 1 | 0 |

**'AND' ('&&'):**

| a | b | a && b |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

**'OR' ('||'):**

| a | b | a \|\| b |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

## Logic 101

Use logic to connect multiple conditions and test for certain cases (0 is false, 1 is true):

**‘NOT’ (‘~’, ’!’):**

| a | !a |
|---|---|
| 0 | 1 |
| 1 | 0 |

**‘AND’ (‘&&’):**

| a | b | a && b |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

**‘OR’ (‘||’):**

| a | b | a \|\| b |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

**‘XOR’:**

| a | b | a xor b |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

## Logic 101

Use logic to connect multiple conditions and test for certain cases (0 is false, 1 is true):

**‘NOT’ (‘~’, ’!’):**

| a | !a |
|---|---|
| 0 | 1 |
| 1 | 0 |

**‘AND’ (‘&&’):**

| a | b | a && b |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

**‘OR’ (‘||’):**

| a | b | a \|\| b |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

**‘XOR’:**

| a | b | a xor b |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

### Examples

- ‘Friday Beer’ if **not** younger than 21 **and** it is Friday.

## Logic 101

Use logic to connect multiple conditions and test for certain cases (0 is false, 1 is true):

**‘NOT’ (‘~’, ’!’):**

| a | !a |
|---|---|
| 0 | 1 |
| 1 | 0 |

**‘AND’ (‘&&’):**

| a | b | a && b |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

**‘OR’ (‘||’):**

| a | b | a \|\| b |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

**‘XOR’:**

| a | b | a xor b |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

### Examples

- ‘Friday Beer’ if **not** younger than 21 **and** it is Friday.
- ‘Discard data’ if outlier **or** affected by unmodeled processes.