

Python

Python 3.2:

/net/local/bin/python3.2

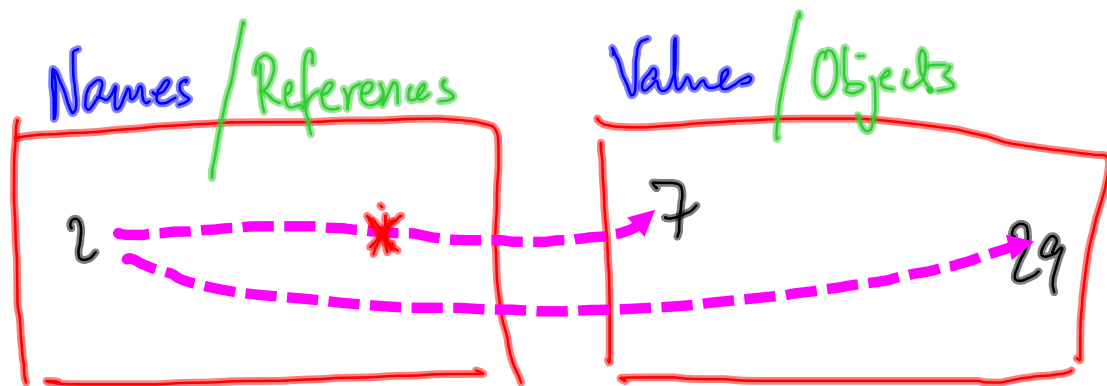
Edit .bashrc

Add a line

```
export PATH=$PATH:/net/local/bin
```

↑↑
no spaces

Names & Values

 $i = 7$ $i = i + 22$

Simple Values

Numbers integers, real numbers

7.2 8 6.39

$\sqrt{16}$

Arithmetic operations

+ , - , * , /

%

mod = remainder

More functions sqrt, sin, cos, ..
in math module

//

quotient (integer)

from math import *

**

exponent

Boolean True, False
 and not or
 not(True) = False

Relational operators that result in Booleans

$x == e$

$x != e$ not equal to not($x == e$)

$x < e$, $x > e$, $x <= e$, $x >= e$

Using Booleans

```
def even(n):  
    if n%2 == 0:  
        return True  
    else:  
        return False
```

```
def even(n):  
    =
```

```
    .  
    .  
    .
```

```
    if even(x):  
        =  
    else:  
        =
```

```
def divides(m,n):  
    if n%m == 0:  
        return True  
    else  
        return False
```

$$m \mid n$$

```
def gcd(m,n):  
    if divides(n,m):  
        return n  
    else:  
        return gcd(n, m%n)
```

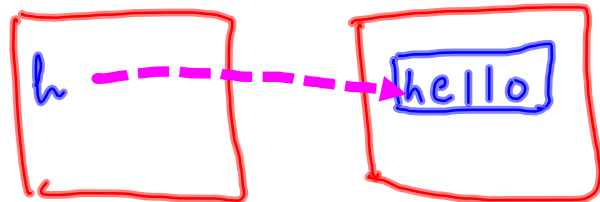
Normally

Expect a single character to be a simple value

Python directly has strings

Sequences of characters

`h = "hello"`



Defining strings

"hello"

'world'

phrase = 'He said "Hello!"'

headline = "Manmohan's folly"

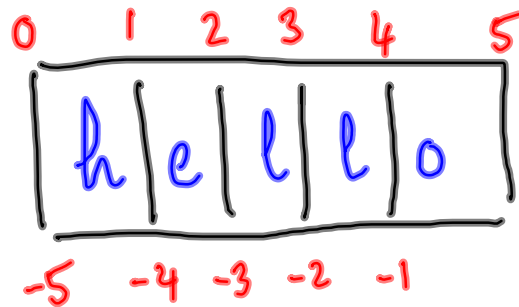
'''
'''

'''
'''

} Triple quote


```
longtext = """ This is a  
very long  
line in Python """
```

Sequence



```
msg = "hello"  
msg[0] - "h"  
msg[4] - "o"
```

msg = "hello" -----> "hell!"
msg[4] = "!" x Not allowed

Strings are **IMMUTABLE**

- A value cannot be changed
- Changing a value produces a new value

Slice

msg = "hello world"

0 1 2

msg[2:5] → ell

start position end position

0 1 2 3 4 5 6 7 8 9 10
hello world
0 1 2 3 4 5 6 7 8 9 10 11

Concatenating strings

Putting together

+ "hello" + " " + "world" → "hello world"

msg = "hello" → msg = "help!"

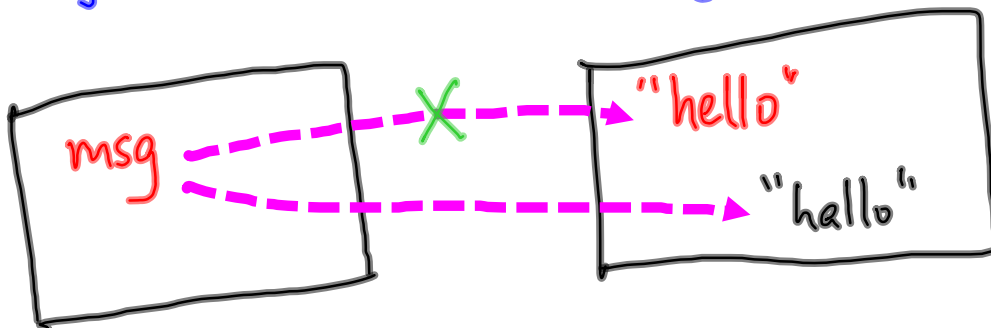
msg = msg[0:3] + "p!"

`msg = "hello"` \dashrightarrow `msg = "hallo"`

`msg = msg[0:1] + "a" + msg[2:5]`

OR

`msg = msg[0] + "a" + msg[2:5]`



```
print(msg[0] + msg[2:5]) --> "hello"
```

`str(x)` Converts `x` to a string

`string * n` Makes `n` copies of string

Sequences of arbitrary values

List

$l = [7, 2.6, True, \text{"hello"}]$

$l[1] \dashrightarrow 2.6$

$l[0:3] \dashrightarrow [7, 2.6, True]$

`l = [[1,2,3], 4, [6], "hello"]`

`l[0][1:3]`



`[1,2,3] [1:3]`



`[2,3]`

`l[3][4]`



`"hello" [4] → "o"`

$l[i]$ i out of range Error

$l[i:j]$ i is too small $l=0$
 l is too large $l = \text{length } l$

$l[i:]$ \equiv $l[i: \text{length of } l]$

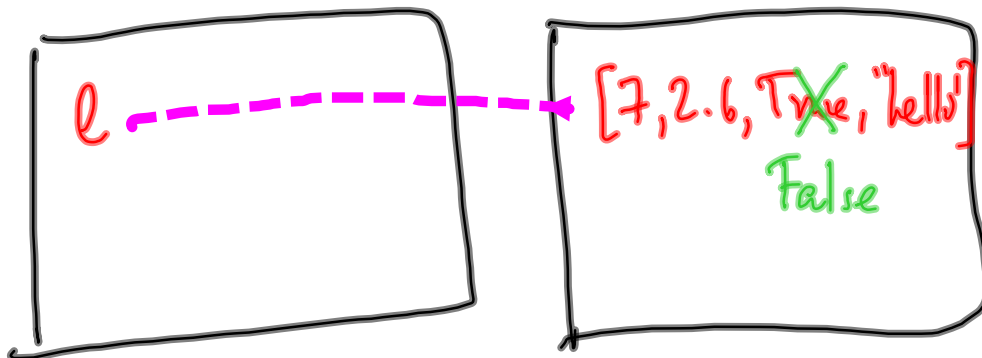
$l[:i]$ \equiv $l[0:i]$

Full slice $l[0: \text{len}(l)] \equiv l[:]$
 \hookrightarrow length of l

Lists are **MUTABLE**

```
l = [7, 2.6, True, "hello"]
```

```
l[2] = False
```

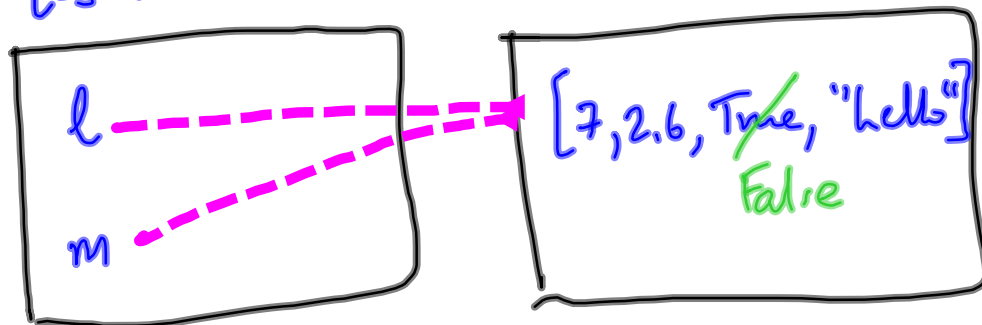


$l = [7, 2.6, True, "hello"]$

$m = l$

$l[2] = False$

$\therefore m[2] ? \rightarrow False !!$



Making a copy of a list
Take a "full" slice

$m = l[0:\text{len}(l)]$

$m = l[:]$

