DSC 10, Spring 2018
Lecture 11

Conditionals and Iteration

sites.google.com/eng.ucsd.edu/dsc-10-spring-2018

Credit: Anindita Adhikari and John DeNero

Random Selection

(Demo)

Random Selection

`np.random.choice`

- Selects at random
- with replacement
- from an array
- a specified number of times

```
np.random.choice(some_array, sample_size)
```

Discussion Question

```
d = np.arange(6) + 1
```

What happens when we evaluate the following 2 expressions?

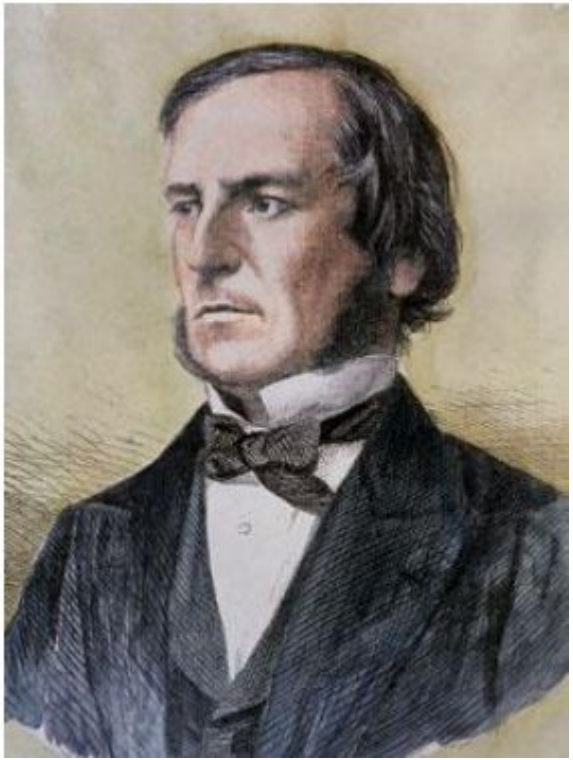
- `np.random.choice(d, 1000) + np.random.choice(d, 1000)`
- `2 * np.random.choice(d, 1000)`

A. Gives the same result; Describing the same process
B. Gives the same result; Describing different processes
C. Gives different results; Describing the same process
D. Gives different results; Describing different processes
E. None of the above

Comparison

George Boole

The Laws of Thought (1854)



Boolean (type)

- “yes/no”, “on/off”
  - Analogy in English
- True, False
- 1, 0

Often the result of a “comparison”:

➔ John has more money than me.

➔ I have more friends than John.

(Demo)

Comparison Operators

The result of a comparison expression is a **`bool`** value

```
x = 2                    y = 3
```

Assignment statements

```
x > 1                    x > y                    y >= 3

x == y                   x != 2                   2 < x < 5
```

Comparison expressions

Combining Comparisons

Boolean operators can be applied to **`bool`** values

```
a = True        b = False
```

Evaluate to True

```
not b           a or b              a and not b
```

```
a and b         not (a or b)        b and b
```

Evaluate to False

- **`or`** : need at least one to be true
- **`and`** : need both to be true

Discussion Question

```
a = True
b = True
not(((not a) and b) or ((not b) or a) )
```

What does the expression evaluate to?

A. True
B. False
C. I'm lost

(Demo)

Conditional Statement

```
  if <if expression>:
     <if body>
elif <elif expression 0>:
     <elif body 0>
elif <elif expression 1>:
     <elif body 1>
...
else:
     <else body>
```

Discussion Question

```
def func(a, b):
    if (a + b > 4 and b > 0):
        return 'print 1'
    elif (a*b >= 4 or b < 0):
        return 'print 2'
    else:
        return 'print 3'
```

What is returned when *func(2, 2)* is called?

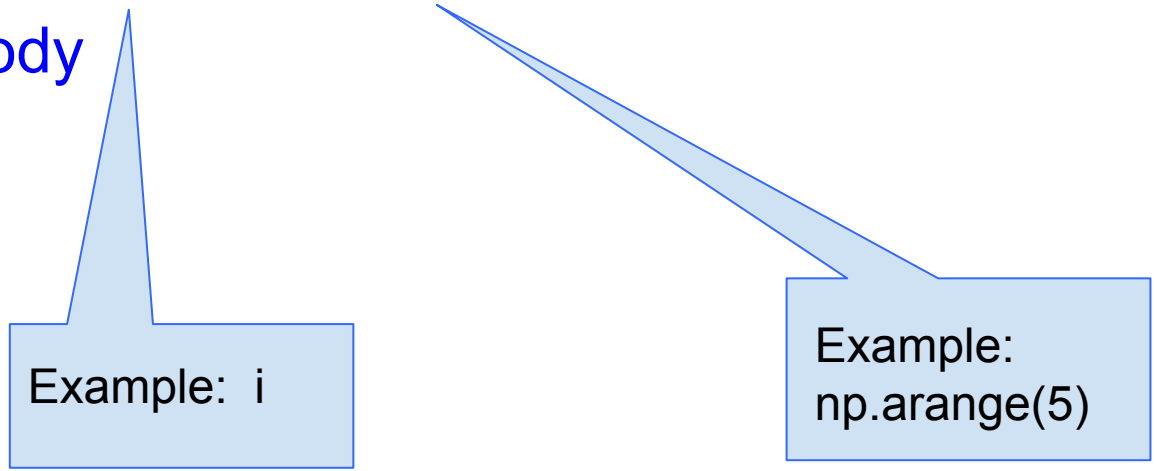
A. print 1
B. print 2
C. print 3
D. More than one print statement
E. Error

(Demo)

For Statement (Loop)

for *item_name* in sequence:

    body



(Demo)

Control Statements

Control Statements

These statements *control* the sequence of computations that are performed in a program

- The keywords `if` and `for` begin control statements
- The purpose of `if` is to define functions that choose different behavior based on their arguments
- The purpose of `for` is to perform a computation for every element in a list or array