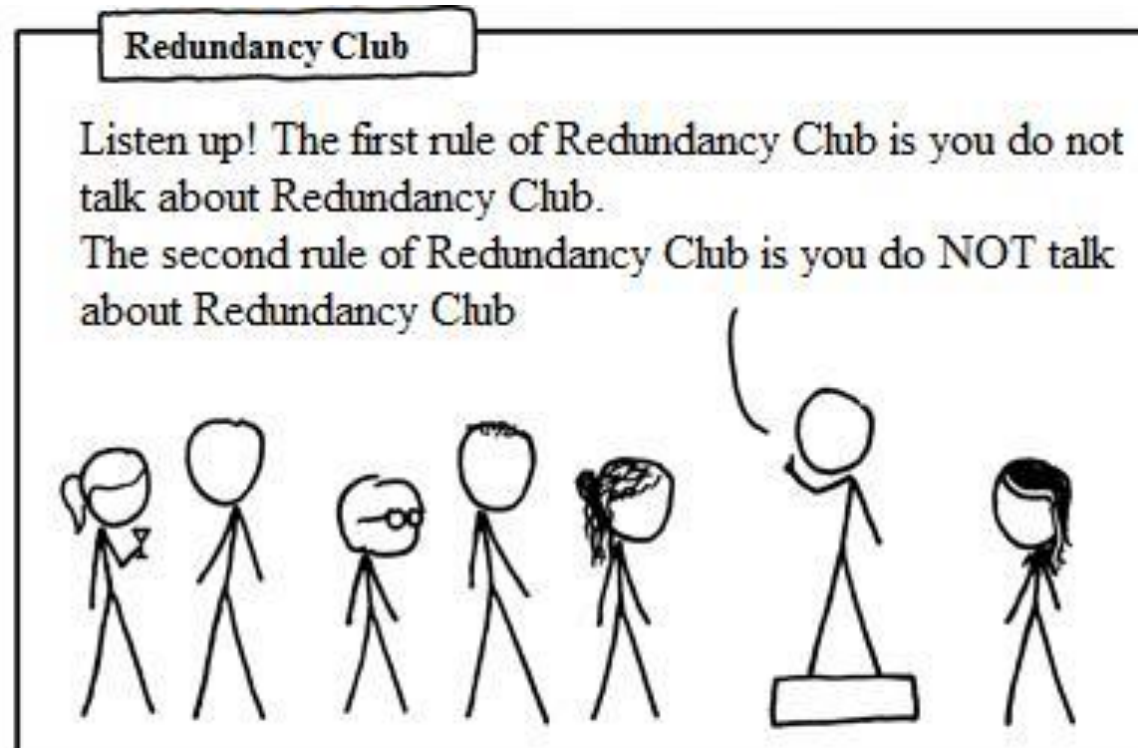# CSc 110, Spring 2018

Lecture 8: `input` and Parameters

Adapted from slides by Marty Stepp and Stuart Reges

Interactive programs

interactive program: Reads input from the console.

- While the program runs, it asks the user to type input.
- The input typed by the user is stored in variables in the code.

- Can be tricky; users are unpredictable and misbehave.
- But interactive programs have more interesting behavior.

input

- **`input`**: An function that can read input from the user.

- Using an `input` object to read console input:

  name = `input`(**prompt**)

  - Example:
    `name = input("type your name: ")`

    - The variable `name` will store the value the user typed in

input example

```
def main():
    age = input("How old are you? ")

    years = 65 - age
    print(years, "years until retirement!")
```

age | 29

- Console (user input underlined):

```
How old are you? 29
```

```
Traceback (most recent call last):
  File "<pyshell#13>", line 1, in <module>
    print(65 - age)
TypeError: unsupported operand type(s) for -:
'int' and 'str'
```

input example

```
def main():
    age = int(input("How old are you? "))

    years = 65 - age
    print(years, "years until retirement!")
```
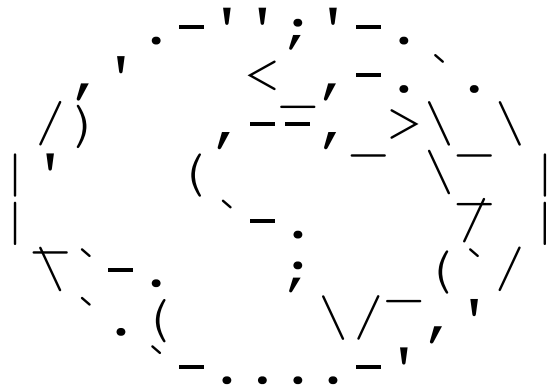
age `29`

years `36`

- Console (user input underlined):

```
How old are you? 29
36 years until retirement!
```

Redundant figures

- Consider the task of printing the following picture:

A redundant solution

```
def main():
    world()
    turtle_equal()
    turtle_v()

def world():
    print("      .-'';'-.")
    print("    ,'    <_,-.`.")
    print("   /)    ,--,_>\\_\\")
    print("  |'    (     \\_ |")
    print("  |      `-.   / |")
    print("   \`-.    ;  (`/")
    print("    .(    \/_,'")
    print("     `-....-'")

def turtle_equal():
    print("              ")
    print("       /= =\\7')")
    print("      /= = =\\/")
    print("        ou ou")

def turtle_v():
    print("              ")
    print("       /v v\\7')")
    print("      /v v v\\/")
    print("        ou ou")

main()
```
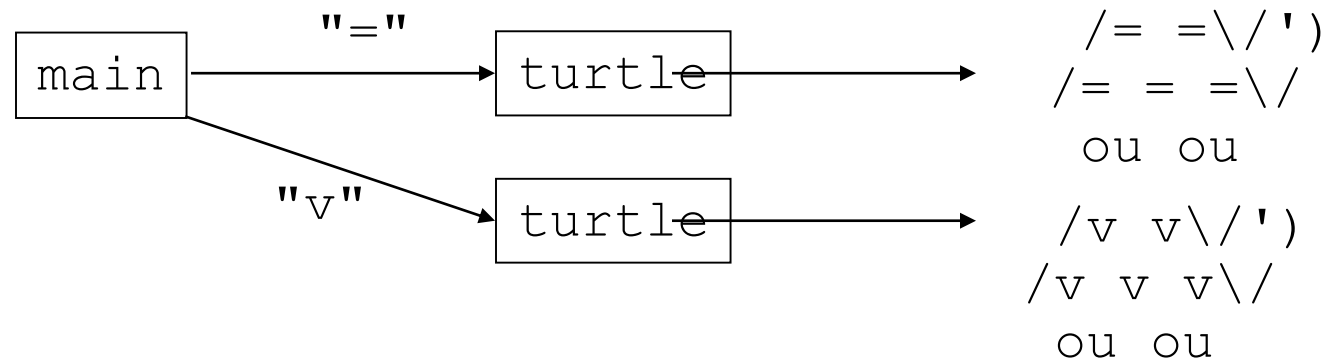
- This code is redundant.

- Would variables help?
  Would constants help?

- What is a better solution?

  - `turtle` - A function to draw a turtle of any shell pattern.

Parameterization

- **parameter**: A value passed to a function by its caller.

  - Instead of `turtle_equal`, `turtle_v`, write `turtle` to draw any turtle.
    - When *declaring* the function, we will state that it requires a parameter for the number of stars.
    - When *calling* the function, we will specify how many stars to draw.

```
            "="                            /= =\/')
┌──────┐         ┌────────┐              /= = =\/
│ main ├────────►│ turtle ├─────────►      ou ou
└──┬───┘         └────────┘
   │      "v"    ┌────────┐              /v v\/')
   └────────────►│ turtle ├─────────►   /v v v\/
                 └────────┘                ou ou
```

Declaring a parameter

Stating that a function requires a parameter in order to run

```
def <name> (<name>):
    <statement>(s)
```

- Example:
```
def say_password(code):
    print("The password is:", code)
```

- When `say_password` is called, the caller must specify the code to print.

Passing a parameter

Calling a function and specifying values for its parameters

<name> (***<expression>***)

- Example:

```
say_password(42)
say_password(12345)
```

Output:

```
The password is 42
The password is 12345
```

Parameters and loops

- A parameter can guide the number of repetitions of a loop.

```
chant(3)
```

```
def chant(times):
    for i in range(0, times):
        print("Just a salad...")
```
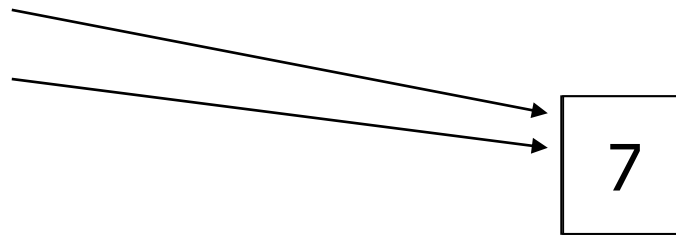
Output:
```
Just a salad...
Just a salad...
Just a salad...
```

How parameters are passed

- When the function is called:
  - The value is stored into the parameter variable.
  - The function's code executes using that value.

```
chant(3)
chant(7)
```

<div style="border:1px solid black; display:inline-block; padding:0 20px;">7</div>

```
def chant(times):
    for i in range(0, times):
        print("Just a salad...")
```

Common errors

- If a function accepts a parameter, it is illegal to call it without passing any value for that parameter.

```
chant()              # ERROR: parameter value required
```

- The value passed to a function must be of a type that will work.

```
chant(3.7)           # ERROR: must be of type int if it
                     #          is used as a range bound
```

- Exercise: Change the `counts` program to use a parameterized function for drawing lines of numbers.

Multiple parameters

- A function can accept multiple parameters. (separate by ，)
  - When calling it, you must pass values for each parameter.

- Declaration:

  ```
  def <name>(<name>, …, <name>):
      <statement>(s)
  ```

- Call:

  <name>(**<exp>**, **<exp>**, …, **<exp>**)

Multiple parameters example

```python
def main():
    print_number(4, 9)
    print_number(17, 6)
    print_number(8, 0)
    print_number(0, 8)

def print_number(number, count):
    for i in range(0, count):
        print(number, end="")
    print()
```

Output:

```
444444444
171717171717

00000000
```

A "Parameter Mystery" problem

```
def main():
    x = 9
    y = 2
    z = 5

    mystery(z, y, x)

    mystery(y, x, z)



def mystery(x, z, y):
    print(z, "and", (y - x))
```

☐ ☐ ☐