ECS15

for and sequence
Comments

- Another way to repeat.
While loop example

count = 1

while count < 11:
    print (count)
    count = count+1
for loop

for count in range(1,11):
    print (count)

- Shorter and sweeter, but exactly the same effect
- range(start,stop) is a built-in function
- Produces the sequence [1,2,...,9,10]
Counting forward

for count in range(0,50,5):
    print (i, end=' ')
Counting backward

for count in range(10, 1, -1):
    print (i)
Using a string

word = "my string"

for char in word:
    print (char, end=' ')
Sequences

- A sequence is an ordered collection of objects.
- Printed enclosed in brackets (the brackets are not part of the sequence)
- Examples:
  - [5, 7, 1, 3] – no reason they have to be in numerical order
  - [“r”, “p”, “s”] – a sequence of one-letter strings
“r” in [“r”, “p”, “s”]

- Check if something is in the sequence: “r” in [“r”, “p”, “s”] is a Boolean expression, and its value is **True**.
- “x” in [“r”, “p”, “s”] is a Boolean expression, and its value is **False**.
Strings are sequences

- A string is a sequence of letters.
- Well, really, a string is a sequence of **characters**.
- Letters, digits, punctuation marks, spaces, are all characters.
- ‘ecs15-f08’ is a sequence of characters.
- ['c', 'o', 'w'] is a sequence of strings, each one character long.
Checking numerical input

```python
x = input("Enter interest rate: ")
# x is a string
```

- Sometimes we want to convert `x` to a number using `int(x)` or `float(x)`
- If `x` is not a string which is a valid integer or floating point number, this causes an error!
Check each digit

- Look at them one by one
- Is each one a digit?

```python
if y in '0123456789':
```
goodInput = False # So far, no valid input
while not goodInput:
    x = input("Enter an integer: ")
goodInput = True
for char in x:
    if char not in '0123456789':
        goodInput = False
if not goodInput:
    print ("That was not an integer!"")