Loops

Presenter: Steve Baskauf
steve.baskauf@vanderbilt.edu

DISC DIGITAL SCHOLARSHIP AND COMMUNICATIONS

Jean & Alexander Heard LIBRARIES

CodeGraf landing page

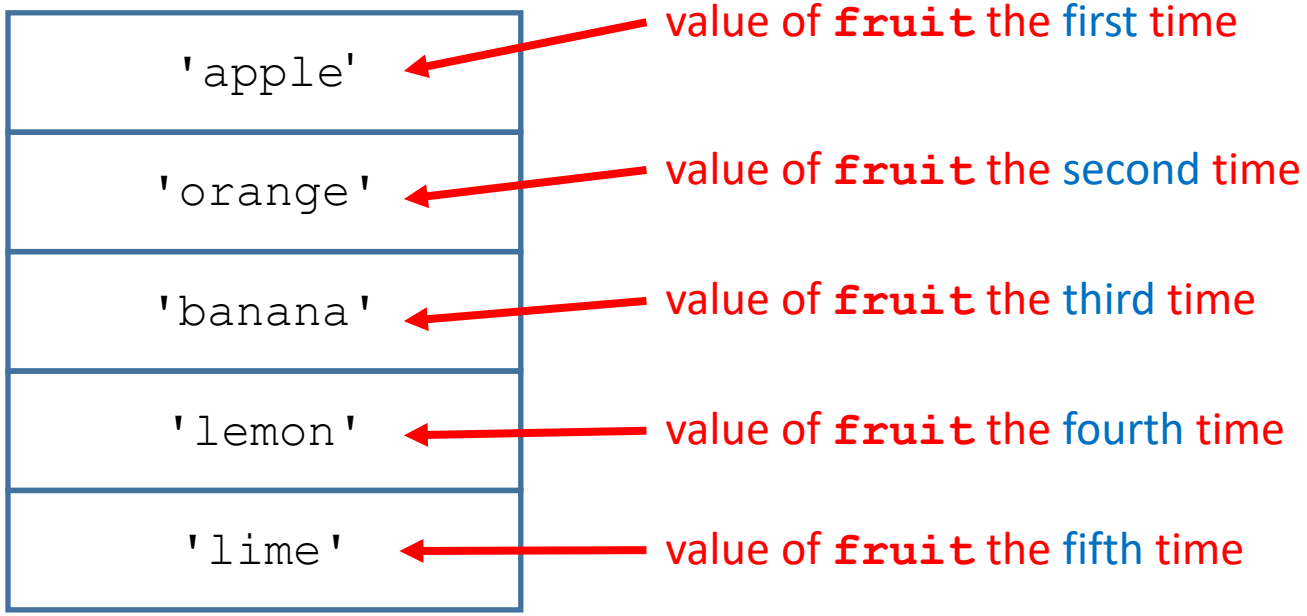
- vanderbi.lt/codegraf

`for` loop

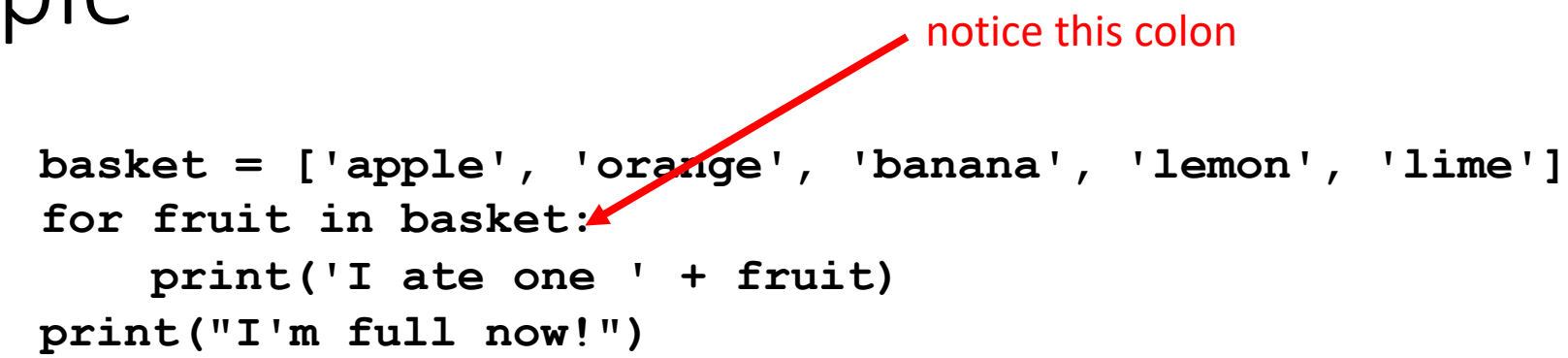
Jean & Alexander Heard
LIBRARIES

Iterating with **`for`**

```
for fruit in basket:
    do this indented code block once for each fruit
then do this code block
```



Example




```
basket = ['apple', 'orange', 'banana', 'lemon', 'lime']
for fruit in basket:
    print('I ate one ' + fruit)
print("I'm full now!")
```

- The indented code block can have more than one line.
- The upcoming code block is signaled by a colon (:)
- Strings are iterable by character.
- **`for`** is useful when there are a definite number of loops

`range()` as an iterable

- The range iterates from the first number to one step less than the second number:
  - `range(1, 11)` iterates from 1 to 10
- A step is optional:
  - `range(2, 10, 2)` iterates by twos from 2 to 8
- The step can be negative:
  - `range(10, 0, -1)` iterates from 10 to 1

Using the value of the range

```
for number in range(1, 11):
    the_square = number**2
    the_area = the_square * 3.14159
    print(number, '\t', the_area)
print("Those are the areas of all the circles!")
```

- The value of the iterated variable can be used anywhere in the indented code block.

Examples

- It's very common to use the length of a list as the end of a range (see last example).
  - Using the length of the list iterates through the whole list because counting is zero-based.

`while` loop

Jean & Alexander Heard
LIBRARIES

Looping with **`while`**

```
power = 0
exponent = 0
print('exponent\tpower')
while power < 100:
    power = 2**exponent
    exponent += 1
    print(exponent, '\t', power)
print("Those are the powers of two.")
```

- **`while`** loops are useful for an indefinite number of loops
- The test value must have an initial value.
- The test value must be able to meet the condition.
- The test is not made again until the loop end

Stringing together methods

DISC DIGITAL SCHOLARSHIP AND COMMUNICATIONS

Jean & Alexander Heard LIBRARIES

Applying methods sequentially

- Recall that functions can be nested inside functions.
- A method can be added onto a method.
- The output of the first method must be the correct type for the second method.
- Example:

```
from datetime import date
this_day = date.today().weekday()
```

Example using strings, lists, and numbers

- Stringing together methods makes compact code
- It also makes less readable code.
- Similar problem to nesting many functions:

```
sqrt(int(input('How many? ')))
```

Remote Support for Teaching and Research Needs

- Access to digital collections 24/7
- Skype consultations with your subject librarian
- Ask a Librarian: an easy way to submit a question via email
- Live chat available from the Library home page

Jean & Alexander Heard
LIBRARIES

NEED HELP? ASK A LIBRARIAN!

https://www.library.vanderbilt.edu/ask-librarian.php

Jean & Alexander Heard
LIBRARIES