

Docstrings

Anonymous Question

Why do we get 3 instead of 7
when we do `-17%10` ?

That is the mathematic definition
of mod (indicated by `%` in
python).

AI

- You should be able to finish this weekend if you want
- DO start now

Docstrings

- If the first thing in a function is a string that isn't assigned to anything, it is a docstring.
- Python saves it so that it can give you help on that function later.
- The name is short for “documentation string.”
- Convention: leave a blank line after (but not before) your docstrings.
- Every function you write should have a docstring.

Why docstrings matter

Reason 1: If you write a docstring before you write the function, you will know what the function is supposed to do!

Many bugs are due to failure to think this through.

Reason 2: If you write a good docstring, another programmer who calls your function knows everything they need to use it properly.

Important since (a) old code lives a long time and (b) new code rarely written from scratch.

Rules for writing great docstrings

- Since docstrings are critically important, you need to write very good ones.
- Here are some principles to follow ...

I. Describe precisely what the function does

```
def vowels(word):
```

```
    """Returns whether the word has vowels."""
```

```
def remove_duplicates(s):
```

```
    """Removes duplicate characters from s."""
```

I. Describe precisely what the function does

```
def vowels(word):
```

```
    """Returns whether the word has vowels."""
```

- True if there are vowels or False?

```
def remove_duplicates(s):
```

```
    """Removes duplicate characters from s."""
```

- “aabcadce” yields “abcde” or “badce”?

Better docstrings

```
def vowels(word):
```

```
    """Return True iff string word has any  
    vowels. Do not treat 'y' as a vowel."""
```

```
def remove_duplicates(s):
```

```
    """Return a new string that is the same as  
    string s, except that only the first  
    occurrence of each character is kept."""
```

2. Do not reveal *how* the function does it.

```
def add_border(pic, c):  
    """Add a border to pic by adding 4  
    overlapping rectangles."""
```

2. Do not reveal *how* the function does it.

```
def add_border(pic, c):
```

```
    """Add a border to pic by adding 4  
    overlapping rectangles."""
```

3. Make the purpose of every parameter clear.

```
def add_border(pic, c):  
    """Add a border to pic."""
```

3. Make the purpose of every parameter clear.

```
def add_border(pic, c):  
    """Add a border to pic."""
```

Better docstring

```
def add_border(pic, c):  
    """Add a 20 pixel wide border of Color  
    c to Picture pic."""
```

4. Refer to every parameter by name.

```
def average_red(pic):
```

```
    """Compute the average amount of red  
    in a picture."""
```

4. Refer to every parameter by name.

```
def average_red(pic):
```

```
    """Compute the average amount of red  
    in a picture."""
```


better docstring

```
def average_red(pic):
```

```
    """Compute the average amount of red  
    in picture pic."""
```

5. Be clear about whether the function returns a value (and if so, what)

```
def average_red(pic):
```

```
    """Compute the average amount of red  
    in a picture."""
```

5. Be clear about whether the function returns a value (and if so, what)

```
def average_red(pic):
```

```
    """Compute the average amount of red  
    in a picture."""
```

- Is the average returned? What type is it?

better docstring

```
def average_red(pic):
```

```
    """Return the average amount of red (a  
    float) in the pixels of Picture pic."""
```

6. Explain any conditions that the function assumes are true.

```
def speed(distance, time):  
    """Return a car's speed."""
```

6. Explain any conditions that the function assumes are true.

```
def speed(distance, time):
```

```
    """Return a car's speed."""
```

- Does the function assume that time must not be zero? Negative?

better docstring

```
def speed(distance, time):
```

```
    """Return the speed (a float) of an object  
    that travels the given distance (an int) in  
    the given time (an int)."""
```

even better

The phrase “the given” is awkward, especially when there are many parameters. But saying “distance distance” is confusing! We sometimes rename parameters so that we can say “distance d” instead.

Also, what about the units?!

```
def speed(d, t):
```

```
    """Return the speed in km/hr (a float) of an  
    object that travels distance d km in  
    time t minutes. d and t are ints."""
```


7. Be concise and grammatically correct.

- I hope this one is self-evident.

8. Write as a command rather than a statement.

```
def cube(x):
```

```
    """Returns the cube of x."""
```

better docstring

```
def cube(x):  
    """Return the cube of x."""
```

Docstring vs “specification”

- On assignments, we often describe functions you are to implement.
- We often provide extra information, such as:
 - example function calls and their results,
 - hints, or
 - the technique to use.
- This is part of the specification of what we want you to do.
- But it does not belong in a docstring.

Example from an old assignment

```
long_side(number)
```

Given the length of the shorter sides of an isosceles right triangle, return the length of the longer side (as a float). An isosceles right triangle has two equal-length sides. The third side is longer and has length equal to the length of one of the other sides multiplied by the square root of 2. (Module math defines a function called sqrt that will be helpful.)

`long_side(number)`

Given the length of the shorter sides of an isosceles right triangle, return the length of the longer side (as a float). An isosceles right triangle has two equal-length sides. The third side is longer and has length equal to the length of one of the other sides multiplied by the square root of 2. (Module `math` defines a function called `sqrt` that will be helpful.)

Docstrings for boolean functions

- Example:
def is_odd(n):
 """Return True if integer n is odd, and False
 otherwise."""
- We often use iff to shorten the docstring:
 """Return True iff integer n is odd."""

- From math: $A \text{ iff } B$ means:
($A \text{ if } B$) and ($A \text{ only if } B$)
- So the docstring above means:
Return true if integer n is odd, and
Return true only if integer n is odd.
(i.e., not under any other circumstances)
- This tells you what to return if n is odd
(return True).
- But not what to return if n is not odd
(it only says *not* to return True!)
- It is understood by programmers that the
function should return False in that case.