LECTURE 17

COMMENTS, DOCUMENTATION, ETC.

MCS 260 Fall 2020
Emily Dumas

REMINDERS

- Quiz 6 due Monday
- Project 2 due Friday Oct 9
- Use the autograder early and often

CONTINUE

This is a loose end from our discussion of loops in Lecture 7.

We talked about `break`, which exits the loop immediately.

There is also `continue`, which skips the rest of the loop body and starts the next iteration.

```python
print(":",end="")
for i in range(10):
    if i in [3,4]:
        continue
    print(i,end="")
    if i == 7:
        break
print(":")
```

has output

```
:012567:
```

More realistic example: Add some error handling to the calculator program from Lecture 15.

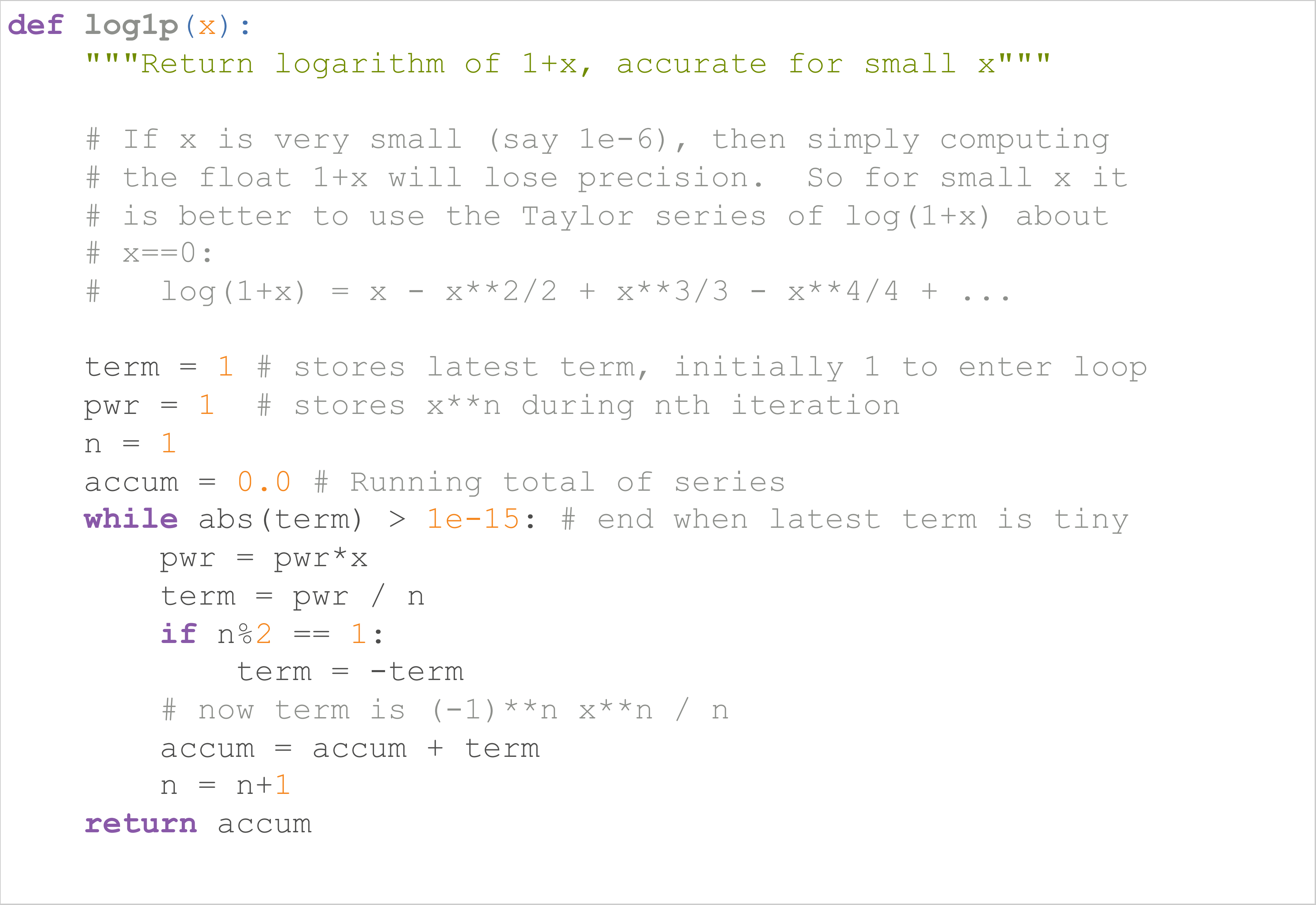
- Code - 10am lecture
- Code - 2pm lecture

COMMENTS

Recall: # begins a comment in Python; rest of line is ignored by interpreter.

Comments exist to help humans, to make code easier to understand.

Docstrings serve similar function, but only exist at top of function or file. Unlike comments, Python remembers docstrings and will print them on request.



```
def log1p(x):
    """Return logarithm of 1+x, accurate for small x"""

    # If x is very small (say 1e-6), then simply computing
    # the float 1+x will lose precision.  So for small x it
    # is better to use the Taylor series of log(1+x) about
    # x==0:
    #   log(1+x) = x - x**2/2 + x**3/3 - x**4/4 + ...

    term = 1 # stores latest term, initially 1 to enter loop
    pwr = 1  # stores x**n during nth iteration
    n = 1
    accum = 0.0 # Running total of series
    while abs(term) > 1e-15: # end when latest term is tiny
        pwr = pwr*x
        term = pwr / n
        if n%2 == 1:
            term = -term
        # now term is (-1)**n x**n / n
        accum = accum + term
        n = n+1
    return accum
```

GOOD COMMENTS

- Clarify the intent (human-readable).
- Explain a key property that holds at a certain point.
- Preview the method or algorithm to come.

BAD COMMENTS

- Duplicate explanation of the obvious:

```python
x = x + 1  # increase x by 1
```

- Substitute for good variable names:

```python
# iterate over items in shopping cart (stored in list c)
for i in c:
    # ...
```

- Out of sync with the code:

```python
# Ban user if they exceed the 3 login attempts
# (the max allowed by our policy)
if attempts > 5 and not is_corporate_network(userip):
    apply_ban(username,14*24*60*60)
```

HOW TO TELL?

Show your code to a classmate (best), or a TA or instructor and ask what is unclear. Add comments there.

DOCUMENTATION

Text for humans that helps make your program more useful to new users, users wanting to know something, or developers.

Docstrings and comments are one type of documentation targeted at developers.

Another important type is documents distributed with your program that describe its operation. Should be formal writing appropriate to your audience.

SOME TYPES OF DOCUMENTATION

- **README** - Meant to be first read; summary of install instructions, basic operation, license, contributors, contact info.
- **Tutorial** - Detailed guide for new users explaining steps to accomplish certain tasks.
- **Reference** - Full technical documentation, often terse, assumes familiarity.

- **README** - This is a unary calculator developed in MCS 260. No installation is needed; run it with the command: ... Here is a sample session where we calculate $2 * 5^3$...
- **Tutorial** - First steps: Running and exiting... Next: Basic calculations...
- **Reference** - Alphabetical list of commands and their functions.

SMALL PROJECT STRUCTURE

```
README.txt
banana.py
...
```

LARGER PROJECT STRUCTURE

```
README.txt
banana/
banana/banana.py
banana/util.py
...
docs/
docs/tutorial.txt
docs/reference.txt
...
```

DOC FORMATS

- **Markdown** (.md) - Text-based format (readable in notepad etc.) that allows for sections, tables, links, code blocks, etc.
- **HTML** (.html) - Meant to be opened in a browser.
- **Plain text** (.txt) - Anyone who can view code can read it, but it can suffer from lack of structure (no sections, headings, etc.) and features (e.g. no links).
- **PDF** (.pdf) - Usually exported from another source format, harder to copy/paste from, not editable.

REFERENCES

- Documentation is discussed in *Section 7.7 of Brookshear & Brylow*.
- This markdown guide is a good starting point for beginners.

REVISION HISTORY

- 2020-10-01 Initial publication