Python Formatting:

f-strings

Marquette University 2019
Why another formatting method

- The format method allows very fine-grained control
- But it is verbose
- Python has two type of special strings:
  - r-strings for raw strings: no escapes
  - f-strings for formatting
- Using f-strings results in more compact and readable code
f-strings

- f-strings are defined with a pair of quotation marks preceded immediately by an “f” or “F”

  \[ \text{fstring} = f\'\text{hello world}' \]

- An f-string can contain a variable name surrounded by brackets in its definition

- The bracket is then replaced by the value of the variable
f-strings

• Example:

```python
number = 6.35
astring = "hello"
fstring = f"{astring}, the number is {number}"
```

• Variable fstring is then

```python
'hello, the number is 6.35'
```
f-strings

• The expression in brackets inside an f-string gets evaluated at run time.

• For example, we can say

\[
\text{f"\{2+3*4\}"}
\]

• or

```python
astring = "hello"
string = f"{astring.upper()} World"
```

which evaluates to

```
'HELLO World'
```
r-strings

- Because of their similarity with f-strings, we mention r-strings

- An r-string uses the escape character only as an escape character, so there is no escaping at all

- This is useful for strings containing the backslash such as Windows file names

  \[\text{address} = \text{r"c:\Windows\System32\system.ini"}\]