Variables, Functions and String Formatting
Code Examples HW 2-1, 2-2
Logical Expressions
Comparison Operators

\[ a == b \]

- Comparison operators *compare* the right-hand side and the left-hand side and return *True* or *False*

- Very similar to logical expressions
Comparison Operators

- **equals**: 5 == 5 -> True  "hello" == "hello" -> True
- **not equals**: 5 != 6 -> True  "cat" != "dog" -> True
- **greater than**: 6 > 5 -> True  "help" < "hello" -> False
- **greater than or equal to**: 5 >= 5 -> True  "hello" >= "hello" -> True
- **less than**: 4 < 5 -> True  "hello" < "help" -> True
- **less than or equal to**: 5 <= 5 -> True  "hello" <= "hello" -> True
Comparisons with float values

- Remember float operations are only approximations

- If you want to check if the result of an expression equals a float value, check instead if their difference is small

```python
#!/usr/bin/env python3
a = 0.1 + 0.2
print(a == 0.3)
```

```
$ python3 test_float.py
$ False
```
Comparisons of Floats

\[ a = 0.1 + 0.2 \]
\[ b = 0.3 \]
\[ \text{print}(a == b) \]

\[ \text{print}(a - b < 0.00001) \]

\[ \text{print}(\text{abs}(a - b) < 0.00001) \]

\[ a = 10.0 \times 10^{-10} \]
\[ b = 10.0 \times 10^{-6} \]
\[ \text{print}(\text{abs}(a - b) / a < 0.00001) \]

\[ \text{print}(\text{abs}(a - b) / \text{abs}(\text{max}(a,b)) < 0.00001) \]
Comparisons with float values

• Moral of the story, use Python’s built-in close check

```python
#!/usr/bin/env python3
import math

a = 0.1 + 0.2
print(math.isclose(a, 0.3))
```
Comparisons With Strings

• Strings compare alphabetically

• To check if a substring is contained within a larger string, use in

• For any string comparisons, make sure they in the same case using str.lower()

"i" in "team" -> False
"fun" in "function" -> True

a = "Fun"
b = "I’m having fun with functions"
print(a in b) -> False
print(a.lower() in b.lower()) -> True
not, and, and/or or

- not, and, or are operators

- and: evaluates to True if the left-hand side is True and the right-hand side is True

- or: evaluates to True if either side is True

- not: inverts a True to False or vice versa

- For comparing more than two values, need to use multiple ands/ors

<table>
<thead>
<tr>
<th>Condition</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>True and True</td>
<td>True</td>
</tr>
<tr>
<td>True and False</td>
<td>False</td>
</tr>
<tr>
<td>True or False</td>
<td>True</td>
</tr>
<tr>
<td>False or False</td>
<td>False</td>
</tr>
<tr>
<td>True and not False</td>
<td>True</td>
</tr>
<tr>
<td>“fun” not in “function”</td>
<td>False</td>
</tr>
<tr>
<td>a and b and c</td>
<td>False</td>
</tr>
<tr>
<td>a or b or c</td>
<td>True</td>
</tr>
<tr>
<td>(a or b) and (c or d)</td>
<td>True</td>
</tr>
</tbody>
</table>
Strings and Text
<table>
<thead>
<tr>
<th>Decimal</th>
<th>Hex</th>
<th>ASCII</th>
<th>Decimal</th>
<th>Hex</th>
<th>ASCII</th>
<th>Decimal</th>
<th>Hex</th>
<th>ASCII</th>
<th>Decimal</th>
<th>Hex</th>
<th>ASCII</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>(NULL)</td>
<td>32</td>
<td>20</td>
<td>(SPACE)</td>
<td>64</td>
<td>40</td>
<td>@</td>
<td>96</td>
<td>60</td>
<td>~</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>[START OF HEAD]</td>
<td>33</td>
<td>21</td>
<td>!</td>
<td>65</td>
<td>41</td>
<td>A</td>
<td>97</td>
<td>61</td>
<td>a</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>[START OF TEXT]</td>
<td>34</td>
<td>22</td>
<td>&quot;</td>
<td>66</td>
<td>42</td>
<td>B</td>
<td>98</td>
<td>62</td>
<td>b</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>[END OF TEXT]</td>
<td>35</td>
<td>23</td>
<td>#</td>
<td>67</td>
<td>43</td>
<td>C</td>
<td>99</td>
<td>63</td>
<td>c</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>[END OF TRANSMISSION]</td>
<td>36</td>
<td>24</td>
<td>$</td>
<td>68</td>
<td>44</td>
<td>D</td>
<td>100</td>
<td>64</td>
<td>d</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>[ENQUIRY]</td>
<td>37</td>
<td>25</td>
<td>%</td>
<td>69</td>
<td>45</td>
<td>E</td>
<td>101</td>
<td>65</td>
<td>e</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>[ACKNOWLEDGE]</td>
<td>38</td>
<td>26</td>
<td>&amp;</td>
<td>70</td>
<td>46</td>
<td>F</td>
<td>102</td>
<td>66</td>
<td>f</td>
</tr>
<tr>
<td>7</td>
<td>7</td>
<td>[BEL]</td>
<td>39</td>
<td>27</td>
<td>'</td>
<td>71</td>
<td>47</td>
<td>G</td>
<td>103</td>
<td>67</td>
<td>g</td>
</tr>
<tr>
<td>8</td>
<td>8</td>
<td>[BACKSPACE]</td>
<td>40</td>
<td>28</td>
<td>(</td>
<td>72</td>
<td>48</td>
<td>H</td>
<td>104</td>
<td>68</td>
<td>h</td>
</tr>
<tr>
<td>9</td>
<td>9</td>
<td>[HORIZONTAL TAB]</td>
<td>41</td>
<td>29</td>
<td>)</td>
<td>73</td>
<td>49</td>
<td>i</td>
<td>105</td>
<td>69</td>
<td>i</td>
</tr>
<tr>
<td>11</td>
<td>B</td>
<td>[VERTICAL TAB]</td>
<td>43</td>
<td>2B</td>
<td>+</td>
<td>75</td>
<td>4B</td>
<td>K</td>
<td>107</td>
<td>6B</td>
<td>k</td>
</tr>
<tr>
<td>12</td>
<td>C</td>
<td>[FORM FEED]</td>
<td>44</td>
<td>2C</td>
<td>,</td>
<td>76</td>
<td>4C</td>
<td>L</td>
<td>108</td>
<td>6C</td>
<td>l</td>
</tr>
<tr>
<td>13</td>
<td>D</td>
<td>[CARRIAGE RETURN]</td>
<td>45</td>
<td>2D</td>
<td>-</td>
<td>77</td>
<td>4D</td>
<td>M</td>
<td>109</td>
<td>6D</td>
<td>m</td>
</tr>
<tr>
<td>14</td>
<td>E</td>
<td>[SHIFT OUT]</td>
<td>46</td>
<td>2E</td>
<td>_</td>
<td>78</td>
<td>4E</td>
<td>N</td>
<td>110</td>
<td>6E</td>
<td>n</td>
</tr>
<tr>
<td>15</td>
<td>F</td>
<td>[SHIFT IN]</td>
<td>47</td>
<td>2F</td>
<td>I</td>
<td>79</td>
<td>4F</td>
<td>O</td>
<td>111</td>
<td>6F</td>
<td>o</td>
</tr>
<tr>
<td>16</td>
<td>10</td>
<td>[DATA LINK ESCAPE]</td>
<td>48</td>
<td>30</td>
<td>0</td>
<td>80</td>
<td>50</td>
<td>P</td>
<td>112</td>
<td>70</td>
<td>p</td>
</tr>
<tr>
<td>17</td>
<td>11</td>
<td>[DEVICE CONTROL 1]</td>
<td>49</td>
<td>31</td>
<td>1</td>
<td>81</td>
<td>51</td>
<td>Q</td>
<td>113</td>
<td>71</td>
<td>q</td>
</tr>
<tr>
<td>18</td>
<td>12</td>
<td>[DEVICE CONTROL 2]</td>
<td>50</td>
<td>32</td>
<td>2</td>
<td>82</td>
<td>52</td>
<td>R</td>
<td>114</td>
<td>72</td>
<td>r</td>
</tr>
<tr>
<td>19</td>
<td>13</td>
<td>[DEVICE CONTROL 3]</td>
<td>51</td>
<td>33</td>
<td>3</td>
<td>83</td>
<td>53</td>
<td>S</td>
<td>115</td>
<td>73</td>
<td>s</td>
</tr>
<tr>
<td>20</td>
<td>14</td>
<td>[DEVICE CONTROL 4]</td>
<td>52</td>
<td>34</td>
<td>4</td>
<td>84</td>
<td>54</td>
<td>T</td>
<td>116</td>
<td>74</td>
<td>t</td>
</tr>
<tr>
<td>21</td>
<td>15</td>
<td>[NEGATIVE ACKNOWLEDGE]</td>
<td>53</td>
<td>35</td>
<td>5</td>
<td>85</td>
<td>55</td>
<td>U</td>
<td>117</td>
<td>75</td>
<td>u</td>
</tr>
<tr>
<td>22</td>
<td>16</td>
<td>[SYNCHRONOUS IDLE]</td>
<td>54</td>
<td>36</td>
<td>6</td>
<td>86</td>
<td>56</td>
<td>V</td>
<td>118</td>
<td>76</td>
<td>v</td>
</tr>
<tr>
<td>23</td>
<td>17</td>
<td>[ENG OF TRANS. BLOCK]</td>
<td>55</td>
<td>37</td>
<td>7</td>
<td>87</td>
<td>57</td>
<td>W</td>
<td>119</td>
<td>77</td>
<td>w</td>
</tr>
<tr>
<td>24</td>
<td>18</td>
<td>[CANCEL]</td>
<td>56</td>
<td>38</td>
<td>8</td>
<td>88</td>
<td>58</td>
<td>X</td>
<td>120</td>
<td>78</td>
<td>x</td>
</tr>
<tr>
<td>25</td>
<td>19</td>
<td>[END OF MEDIUM]</td>
<td>57</td>
<td>39</td>
<td>9</td>
<td>89</td>
<td>59</td>
<td>Y</td>
<td>121</td>
<td>79</td>
<td>y</td>
</tr>
<tr>
<td>26</td>
<td>1A</td>
<td>[SUBSTITUTE]</td>
<td>58</td>
<td>3A</td>
<td>:</td>
<td>90</td>
<td>5A</td>
<td>Z</td>
<td>122</td>
<td>7A</td>
<td>z</td>
</tr>
<tr>
<td>27</td>
<td>1B</td>
<td>[ESCAPE]</td>
<td>59</td>
<td>3B</td>
<td>;</td>
<td>91</td>
<td>5B</td>
<td>[</td>
<td>123</td>
<td>7B</td>
<td>{</td>
</tr>
<tr>
<td>28</td>
<td>1C</td>
<td>[FILE SEPARATOR]</td>
<td>60</td>
<td>3C</td>
<td>&lt;</td>
<td>92</td>
<td>5C</td>
<td>}</td>
<td>124</td>
<td>7C</td>
<td>}</td>
</tr>
<tr>
<td>29</td>
<td>1D</td>
<td>[GROUP SEPARATOR]</td>
<td>61</td>
<td>3D</td>
<td>=</td>
<td>93</td>
<td>5D</td>
<td>]</td>
<td>125</td>
<td>7D</td>
<td>]</td>
</tr>
<tr>
<td>30</td>
<td>1E</td>
<td>[RECORD SEPARATOR]</td>
<td>62</td>
<td>3E</td>
<td>&gt;</td>
<td>94</td>
<td>5E</td>
<td>^</td>
<td>126</td>
<td>7E</td>
<td>~</td>
</tr>
<tr>
<td>31</td>
<td>1F</td>
<td>[UNIT SEPARATOR]</td>
<td>63</td>
<td>3F</td>
<td>?</td>
<td>95</td>
<td>5F</td>
<td>_</td>
<td>127</td>
<td>7F</td>
<td>(DEL)</td>
</tr>
</tbody>
</table>
Special Characters

- Some characters in strings have special meanings
- You can tell them apart because they start with a backslash

```
print("He said, \"Hello.\"\")
He said, “Hello.”
```

```
print(‘He didn’t go to the store’) 
He didn’t go to the store
```

```
print(“This text is two lines.\nThis is line 2.”)
This text is two lines.
This is line 2.
```
Methods
Methods vs Functions

• **Methods** are functions that operate on a specific variable or value.

• Syntax is `variable.method_name()`.

• The method performs computations specific to the variable.

```python
name = "Stephen"
print(name.lower())
print(name.upper())
print(name.replace('e', ' '))
```
str.format(param1, param2,...)

• String concatenation is bug prone because all values must be strings

• An incorrect print statement at the end of a long running program is incredibly infuriating

• Calling the method format on your string will do the conversion for you

#!/usr/bin/env python3

name = "Stephen"
age = 32
print("My name is {} and my age is {}".format("Stephen", 32))
str.format(param1, param2,...)

- Place {} in your strings where you want the parameter to go
- Parameters are inserted in order, unless otherwise specified

#!/usr/bin/env python3

name = "Stephen"
age = 32
print("My name is {} and my age is {}".format("Stephen", 32))
Named Parameters

• Functions and methods have variable names associated with the input parameters

• Including a reference to the name of the parameter often results easier to read code

• For str.format() it also allows you to fill in the blanks by name

message = “Hello, my name is {name} and you killed my {relation}”
# Some number of lines of code

my_name = “Inigo Montoya”
relation = “Father”
print(message.format(name = my_name, relation = relation))
To find methods that are available for a specific value type, you can call `dir()` in the interactive mode.

Methods with two underscores before and after it are not meant to be used, they are called **private** methods.

Use `help()` to get more information about a specific method.
Style

- Operations should have spaces around them: $a = b + c$, not $a=b+c$
- Parameters in functions and methods should have spaces after comma `str.format(a, b)`
- Named parameters in functions should be spaced appropriately `str.format(a = "a", b = "b")`