CS 49 Section

Week 6 Bonus Slides

Surajit A Bose

Bitwise Operators: & and |

- In addition to the logical operators **and** and **or**, Python also has bitwise and **&** and bitwise or **|**
- These operate on the binary representations of the operand values
- A binary representation is a representation in base 2
- An explanation of base 2 representation is in this Khan Academy video
- Given two binary numbers, the bitwise operators compare each bit of the left operand to the corresponding bit of the right operand
  - bitwise **&** yields 1 if both the bits are 1, and 0 if either bit is 0
  - bitwise **|** yields 0 if both the bits are 0, and 1 if either bit is 1
- This is analogous to when the logical **and** and **or** return **True** or **False**

Bitwise Operators: & and |

- `1 & 1` `# result: 1`
- `1 & 0` `# result: 0`
- `0 & 1` `# result: 0`
- `0 & 0` `# result: 0`

- `1 | 1` `# result: 1`
- `1 | 0` `# result: 1`
- `0 | 1` `# result: 1`
- `0 | 0` `# result: 0`

Bitwise Operators: & and |

- Example: decimal 5 is binary 101, decimal 2 is binary 10
- For convenience, we can add a leading zero to the latter so the two have the same number of digits: 010
- What is **`5 & 2`**?

```
    101
&   010
-------

    000  # Decimal 0
```

What is **`5 | 2`**?

```
    101
|   010
-------

    111  # Decimal 7
```

- Since no place has a 1 in both binary numbers, the **`&`** evaluates to 0
- Since every place has a 1 in one or the other, the **`|`** evaluates to 7

Bitwise Operators: & and |

```
def bitwise_operations():
    a = 3         # decimal 3: binary 011
    b = 6         # decimal 6: binary 110
    c = a & b     # result: ?
    d = a | b     # result: ?

    print(f'Result of bitwise and, 3 & 6 : {c}')
    print(f'Result of bitwise or,  3 | 6 : {d}')

if __name__ == '__main__':
    bitwise_operations()
```

Bitwise Operators: & and |

```
def bitwise_operations():
    a = 3          # decimal 3: binary 011
    b = 6          # decimal 6: binary 110
    c = a & b      # result:         & 010 decimal 2
    d = a | b      # result:         | 111 decimal 7

    print(f'Result of bitwise and, 3 & 6 : {c}')
    print(f'Result of bitwise or,  3 | 6 : {d}')

if __name__ == '__main__':
    bitwise_operations()
```

That's all, folks!