**Puzzle 1**

((True and False) or (False and (not True))) or (not False**)**

**Puzzle 2**

e = "\\t represents a tab\n\\n represents a new line"

**Puzzle 3**

1. int(True) → 1:
   * True is converted to an integer, which is 1.
2. str(True) → 'True':
   * True is converted to a string, which is 'True'.
3. float(False) → 0.0:
   * False is treated as 0 in numeric contexts, and converting 0 to a float gives 0.0
4. str(False) → 'False':
   * False is converted to a string, which is 'False'.
5. bool(35) → True:
   * Non-zero numbers are considered True in Boolean context.
6. bool('a') → True:
   * Non-empty strings are considered True in Boolean context.
7. bool(None) → False:
   * None is considered False in Boolean context.
8. True + 1 → 2:
   * True is treated as 1 in arithmetic operations, so 1 + 1 = 2.
9. False - 2 → -2:
   * False is treated as 0 in arithmetic operations, so 0 - 2 = -2.
10. True + 'a' → Error:
    * You cannot add a Boolean (True) and a string ('a'), so a TypeError occurs

**Puzzle 4**

1. [4:-1] → "n to progra"

* 4 is 'n', -1 is 'm' (excluded).

1. a[4:len(a)] → "n to program"

* len(a) is 17, so it slices till the end

**Puzzle 5**

**capitalize()** → Converts first character to uppercase, rest to lowercase. Eg "hello world".capitalize() # "Hello world"

**casefold()** → Converts to lowercase (stronger than lower(), handles special cases). Eg "ß".casefold() # "ss"

**lower()** → Converts all characters to lowercase. Eg "Hello".lower() # "hello"

**swapcase()** → Swaps uppercase to lowercase and vice versa. Eg "HeLLo".swapcase() # "hEllO"

**title()** → Capitalizes the first letter of each word. Eg "hello world".title() # "Hello World"

**upper()** → Converts all characters to uppercase. Eg "hello".upper() # "HELLO"

**count(substring)** → Counts occurrences of a substring. Eg "banana".count("a") # 3

**find(substring)** → Returns the first index of a substring, or -1 if not found. Eg "hello".find("l") # 2

**endswith(suffix)** → Checks if string ends with a given suffix. Eg "hello.txt".endswith(".txt") # True

**index(substring)** → Like find(), but raises an error if not found. Eg "hello".index("l") # 2

**startswith(prefix)** → Checks if string starts with a given prefix. Eg "hello".startswith("he") # True

**isalnum()** → Checks if all characters are letters or digits. Eg "Hello123".isalnum() # True

**isalpha()** → Checks if all characters are letters. Eg "Hello".isalpha() # True

**islower()** → Checks if all characters are lowercase. Eg "Hello".isalpha() # True

**isupper()** → Checks if all characters are uppercase. Eg "HELLO".isupper() # True

**isnumeric()** → Checks if all characters are numbers. Eg "123".isnumeric() # True

**replace(old, new)** → Replaces occurrences of old with new. Eg "hello world".replace("world", "Python") # "hello Python"

**strip()** → Removes leading and trailing whitespace (or specified characters. Eg " hello ".strip() # "hello"