|  |  |  |
| --- | --- | --- |
| **Word** | **Name** | **Description** |
| Print | Print | Prints the contents of the brackets.  print("This is fun.")  print("Hens", 25 + 30 / 6)  print(3 + 2 < 5 - 7)  cars = 100  print("There are", cars, "cars available.")  my\_name = 'Zed A. Shaw'  print(f"Let's talk about {my\_name}.")  types\_of\_people = 10  x = f"There are {types\_of\_people} types of people."  print(x)  print(f"I said: {x}")  hilarious = False  joke\_evaluation = "Isn't that joke so funny?! {}"  print(joke\_evaluation.format(hilarious))  print(end1 + end2 + end3 + end4 + end5 + end6, end=' ')  formatter = "{} {} {} {}"  print(formatter.format(1, 2, 3, 4))  days = "Mon Tue Wed Thu Fri Sat Sun"  months = "Jan\nFeb\nMar\nApr\nMay\nJun\nJul\nAug"  print("Here are the days: ", days)  print("Here are the months: ", months)  print("""  There's something going on here.  With the three double-quotes.  We'll be able to type as much as we like.  Even 4 lines if we want, or 5, or 6.  """) |
| Input |  | print("How old are you?", end=' ')  age = input()  print(f"So, you're {age} old, {height} tall and {weight} heavy.")  age = input("How old are you? ")  print(f"So, you're {age} old, {height} tall and {weight} heavy.") |
| argv |  | from sys import argv  script, first, second, third = argv  print("The script is called:", script)  script, user\_name = argv  print(f"Hi {user\_name}, I'm the {script} script.") |
| open |  | script, filename = argv  txt = open(filename)  print(f"Here's your file {filename}:") |
| read |  | txt = open(filename)  print(txt.read()) |
| close |  | Closes the file. Like File->Save.. in your editor. |
| read |  | Reads the contents of the file. You can assign the result to a variable. |
| readline |  | Reads just one line of a text file. |
| truncate |  | Empties the file. Watch out if you care about the file.  target.truncate() |
| write('stuff') |  | Writes ”stuff” to the file.  target.write(line1) |
| seek(0) |  | Move the read/write location to the beginning of the file. |
| close |  | target.close() |
| def | Define Function | def print\_all(f):  print(f.read()) |
| return |  | def add(a, b):  print(f"ADDING {a} + {b}")  return a + b |

|  |  |  |
| --- | --- | --- |
| **Symbol** | **Name** | **Description** |
| # | Pound | Use at the start of a line to make a comment. |
| + | Add |  |
| - | Subtract |  |
| / | Divide |  |
| \* | Multiply |  |
| % | Mod | Find’s the remainder of division of two numbers. |
| < | Less Than |  |
| > | Greater Than |  |
| = | Equal |  |
| <= | Less Than Equal |  |
| >= | Greater Than Equal |  |
| \t | Tab |  |
| \\ | Backslash |  |
| \’ | Single Quote |  |
| \” | Double Quote |  |
| \a | ASCII bell (BEL) |  |
| \b | ASCII backspace (BS) |  |
| \f | ASCII formfeed (FF |  |
| \n | ASCII linefeed (LF) |  |
| \N{Name} |  | Character named name in the Unicode database (Unicode only) |
| \r | Carriage Return |  |
| \t | Horizontal Tab |  |
| \uxxxx |  | Character with 16-bit hex value xxxx |
| \Uxxxxxxxx |  | Character with 32-bit hex value xxxxxxxx |
| \v |  | ASCII vertical tab (VT) |
| \ooo |  | Character with octal value ooo |
| \xhh |  | Character with hex value hh |