Getting Started with PyCharm: Before we begin, make sure you have PyCharm installed and a new Python project open.

1. **Open PyCharm.**
2. **Create a New Project:** If you don't have one, go to `File > New Project`. Choose a location and make sure "New environment using" is set to `Virtualenv`. Click "Create".
3. **Create a New Python File:** In your new project, right-click on the project name in the left panel (usually named after your project) `> New > Python File`. Give it a name like `basic_tasks.py` (or anything you like). This is where you'll type your code.

Part 1: Basic Output and Variables

Task 1: Hello World

Print the message "Hello, Python!" to the console.

Type Python Code:

```
print("Hello, Python!")
```

Expected Output: Hello, Python!

Task 2: Your Name

Store your first name in a variable and print it.

Type Python Code:

```
first_name = "Alice"  # Change "Alice" to your actual name
print(first_name)
```

Expected Output: Alice

Task 3: Your Age

Store your age in a variable and print it.

Type Python Code:

```
my_age = 30  # Change 30 to your actual age
print(my_age)
```

Expected Output: 30

Task 4: Combining Text

Store "Python" in one variable and "Programming" in another. Print them together like "Python Programming".

Type Python Code:

```
word1 = "Python"
word2 = "Programming"
combined_words = word1 + " " + word2
print(combined_words)
```

Expected Output:

```
Python Programming
```

Task 5: Favorite Number

Store your favorite whole number in a variable and print it.

Type Python Code:

```
fav_number = 7
print(fav_number)
```

Expected Output:

```
7
```

Task 6: Decimal Number

Store a decimal number (like 3.14) in a variable and print it.

Type Python Code:

```
pi_value = 3.14159
print(pi_value)
```

Expected Output:

```
3.14159
```

Task 7: True/False

Store `True` in a variable and print it. Then store `False` in another variable and print it.

Type Python Code:

```
is_sunny = True
is_raining = False
print(is_sunny)
print(is_raining)
```

Expected Output:

```
True
False
```

Task 8: Empty Variable

Create a variable and set its value to `None`. Print it.

Type Python Code:

```
no_value = None
print(no_value)
```

Expected Output: None

Part 2: Basic Math Operations

Task 9: Addition

Add two numbers (e.g., 5 and 3) and print the result.

Type Python Code:

```
num1 = 5
num2 = 3
sum_result = num1 + num2
print("The sum is:", sum_result)
```

Expected Output: The sum is: 8

Task 10: Subtraction

Subtract one number from another (e.g., 10 minus 4) and print the result.

Type Python Code:

```
val1 = 10
val2 = 4
difference = val1 - val2
print("The difference is:", difference)
```

Expected Output: The difference is: 6

Task 11: Multiplication

Multiply two numbers (e.g., 6 by 7) and print the result.

Type Python Code:

```
factor1 = 6
factor2 = 7
product = factor1 * factor2
print("The product is:", product)
```

Expected Output: The product is: 42

Task 12: Division

Divide one number by another (e.g., 20 by 5) and print the result.

Type Python Code:

```
dividend = 20
divisor = 5
quotient = dividend / divisor
print("The quotient is:", quotient)
```

Expected Output: 4.0

Task 13: Remainder (Modulo)

Find the remainder when 17 is divided by 5 and print it.

Type Python Code:

```
number = 17
divisor = 5
remainder = number % divisor
print("The remainder is:", remainder)
```

Expected Output: The remainder is: 2

Task 14: Exponents

Calculate 2 raised to the power of 4 and print the result.

Type Python Code:

```
base = 2
power = 4
result = base ** power
print("2 to the power of 4 is:", result)
```

Expected Output: 2 to the power of 4 is: 16

Task 15: Order of Operations

Calculate `(5 + 3) * 2` and print the result.

Type Python Code:

```
calculation_result = (5 + 3) * 2
print("The calculation result is:", calculation_result)
```

Expected Output: The calculation result is: 16

Task 16: Floating Point Division

Divide 7 by 2 and print the result (it should be a decimal).

Type Python Code:

```
val_a = 7
val_b = 2
decimal_division = val_a / val_b
print("7 divided by 2 is:", decimal_division)
```

Expected Output: 7 divided by 2 is: 3.5

Part 3: Strings (Text)

Task 17: String Length

Find out how many characters are in the word "Anaconda" and print the number.

Type Python Code:

```
word = "Anaconda"
word_length = len(word)
print("The length of 'Anaconda' is:", word_length)
```

Expected Output: The length of 'Anaconda' is: 8

Task 18: Uppercase

Convert the word "hello" to all uppercase letters and print it.

Type Python Code:

```
lower_word = "hello"
upper_word = lower_word.upper()
print("Uppercase:", upper_word)
```

Expected Output: Uppercase: HELLO

Task 19: Lowercase

Convert the word "WORLD" to all lowercase letters and print it.

Type Python Code:

```
upper_word_two = "WORLD"
lower_word_two = upper_word_two.lower()
print("Lowercase:", lower_word_two)
```

Expected Output: Lowercase: world

Task 20: Capitalize

Capitalize the first letter of the sentence "this is a sentence." and print it.

Type Python Code:

```
sentence = "this is a sentence."
capitalized_sentence = sentence.capitalize()
print("Capitalized:", capitalized_sentence)
```

Expected Output: Capitalized: This is a sentence.

Task 21: Replace Text

In the sentence "I like cats.", change "cats" to "dogs" and print the new sentence.

Type Python Code:

```
original_sentence = "I like cats."
new_sentence = original_sentence.replace("cats", "dogs")
print("New sentence:", new_sentence)
```

Expected Output: New sentence: I like dogs.

Task 22: Check if Starts With

Check if the word "Python" starts with the letter 'P' and print `True` or `False`.

Type Python Code:

```
program_lang = "Python"
starts_with_p = program_lang.startswith("P")
print("Starts with 'P'?", starts_with_p)
```

Expected Output: Starts with 'P'? True

Task 23: Check if Ends With

Check if the word "Programming" ends with the letters 'ing' and print `True` or `False`.

Type Python Code:

```
word_to_check = "Programming"
ends_with_ing = word_to_check.endswith("ing")
print("Ends with 'ing'?", ends_with_ing)
```

Expected Output: Ends with 'ing'? True

Task 24: Find Text

Find the starting position of the letter 'o' in the word "Hello" and print the number.

Type Python Code:

```
find_word = "Hello"
position_of_o = find_word.find("o")
print("Position of 'o':", position_of_o) # Python counts from 0
```

Expected Output: Position of 'o': 4

Task 25: String Slicing (Part 1)

From the word "Keyboard", get and print only the first three letters.

Type Python Code:

```
full_word = "Keyboard"
first_three = full_word[0:3] # Get characters from index 0 up to (but not including) index 3
print("First three letters:", first_three)
```

Expected Output: First three letters: Key

Task 26: String Slicing (Part 2)

From the word "Monitor", get and print the letters from the 4th position to the end.

Type Python Code:

```
another_word = "Monitor"
# M o n i t o r
# 0 1 2 3 4 5 6 (indices)
from_fourth = another_word[3:] # Start at index 3 (the 4th character) and go to the end
print("From 4th position:", from_fourth)
```

Expected Output: From 4th position: itor

Task 27: Concatenate Strings

Join the string "Data" with the string "Science" and print the result.

Type Python Code:

```
part1 = "Data"
part2 = "Science"
combined = part1 + part2
print("Combined:", combined)
```

Expected Output: Combined: DataScience

Task 28: Repeat String

Print the word "Ha" five times.

Type Python Code:

```
laugh = "Ha"
repeated_laugh = laugh * 5
print("Repeated laugh:", repeated_laugh)
```

Expected Output:

```
Repeated laugh: HaHaHaHaHa
```

Part 4: User Input

Task 29: Get Name

Ask the user "What is your name?" and store their answer in a variable. Then print "Hello, [their name]!".

Type Python Code:

```
user_name = input("What is your name? ")
print("Hello,", user_name + "!")
```

Instructions: When you run this, PyCharm will wait for you to type in the "Run" window.

Example Interaction & Output:

```
What is your name? Bob  <- (You type 'Bob' and press Enter)
Hello, Bob!
```

Task 30: Get Age

Ask the user "How old are you?" and store their answer. Print "You are [their age] years old."

Type Python Code:

```
user_age = input("How old are you? ")
print("You are", user_age, "years old.")
```

Instructions: Type an age when prompted.

Example Interaction & Output:

```
How old are you? 25  <- (You type '25' and press Enter)
You are 25 years old.
```

Task 31: Simple Calculator (Add)

Ask for two numbers, add them, and print the sum. (Remember to convert input to numbers!)

Type Python Code:

```
num_str1 = input("Enter the first number: ")
num_str2 = input("Enter the second number: ")

number1 = int(num_str1) # Converts text to a whole number
number2 = int(num_str2)

total_sum = number1 + number2
print("The sum of your numbers is:", total_sum)
```

Instructions: Type two numbers when prompted.

Example Interaction & Output:

```
Enter the first number: 15  <- (You type '15' and press Enter)
Enter the second number: 7   <- (You type '7' and press Enter)
The sum of your numbers is: 22
```

Part 5: Lists (Collections of Items)

Task 32: Create a List

Create a list of your three favorite fruits and print the whole list.

Type Python Code:

```
 fruits = ["apple", "banana", "cherry"]
print("My favorite fruits:", fruits)
```

Expected Output:

```
My favorite fruits: ['apple', 'banana', 'cherry']
```

Task 33: Access List Item

From your fruit list, print only the second fruit.

Type Python Code:

```
fruits = ["apple", "banana", "cherry"]
print("The second fruit is:", fruits[1]) # Remember Python uses zero-indexing!
```

Expected Output:

```
The second fruit is: banana
```

Task 34: Add to List

Add a new fruit to your list and print the updated list.

Type Python Code:

```
fruits = ["apple", "banana", "cherry"]
fruits.append("orange")
print("Updated fruits list:", fruits)
```

Expected Output:

```
Updated fruits list: ['apple', 'banana', 'cherry', 'orange']
```

Task 35: Remove from List

Remove one fruit from your list and print the updated list.

Type Python Code:

```
fruits = ["apple", "banana", "cherry", "orange"]
fruits.remove("banana")
print("List after removing banana:", fruits)
```

Expected Output:

```
List after removing banana: ['apple', 'cherry', 'orange']
```

Task 36: List Length

Find out how many items are in your fruit list and print the number.

Type Python Code:

```
fruits = ["apple", "cherry", "orange"]
num_fruits = len(fruits)
print("Number of fruits:", num_fruits)
```

Expected Output:

```
Number of fruits: 3
```

Task 37: Check if Item Exists

Check if "apple" is in your fruit list and print `True` or `False`. Also check for "grape".

Type Python Code:

```
fruits = ["apple", "cherry", "orange"]
is_apple_present = "apple" in fruits
print("Is 'apple' in the list?", is_apple_present)

is_grape_present = "grape" in fruits
print("Is 'grape' in the list?", is_grape_present)
```

Expected Output:

```
Is 'apple' in the list? True
Is 'grape' in the list? False
```

Task 38: List of Numbers

Create a list of 5 numbers and print them.

Type Python Code:

```
numbers = [10, 20, 30, 40, 50]
print("List of numbers:", numbers)
```

Expected Output: List of numbers: [10, 20, 30, 40, 50]

Task 39: Sort a List

Sort your list of numbers from smallest to largest and print the sorted list.

Type Python Code:

```
numbers = [50, 10, 40, 20, 30]
numbers.sort()
print("Sorted numbers:", numbers)
```

Expected Output: Sorted numbers: [10, 20, 30, 40, 50]

Task 40: Reverse a List

Reverse the order of items in your fruit list and print the reversed list.

Type Python Code:

```
fruits = ["apple", "banana", "cherry"]
fruits.reverse()
print("Reversed fruits:", fruits)
```

Expected Output: Reversed fruits: ['cherry', 'banana', 'apple']

Part 6: Basic Conditional Logic (If Statements)

Task 41: Check if Equal

Store a number `x = 10`. If `x` is equal to 10, print "x is 10".

Type Python Code:

```
x = 10
if x == 10:
   print("x is 10")
```

Expected Output: x is 10

Task 42: Check if Greater

Store a number `y = 7`. If `y` is greater than 5, print "y is greater than 5".

Type Python Code:

```
y = 7
if y > 5:
   print("y is greater than 5")
```

Expected Output:

```
y is greater than 5
```

Task 43: Check if Less

Store a number `z = 3`. If `z` is less than 5, print "z is less than 5".

Type Python Code:

```
z = 3
if z < 5:
   print("z is less than 5")
```

Expected Output:

```
z is less than 5
```

Task 44: Even or Odd

Get a number from the user. If it's even, print "Even". Otherwise, print "Odd".

Type Python Code:

```
user_num_str = input("Enter a whole number: ")
user_num = int(user_num_str)

if user_num % 2 == 0:
   print("Even")
else:
   print("Odd")
```

Instructions: Type a number when prompted (e.g., 4 or 7).

Example Interaction & Output 1 (Even):

```
Enter a whole number: 4
Even
```

Example Interaction & Output 2 (Odd):

```
Enter a whole number: 7
Odd
```

Task 45: Positive, Negative, or Zero

Get a number from the user. Print "Positive" if it's greater than 0, "Negative" if less than 0, and "Zero" if it's 0.

Type Python Code:

```
num_input_str = input("Enter any number: ")
number_check = float(num_input_str)

if number_check > 0:
   print("Positive")
elif number_check < 0:
   print("Negative")
else:
   print("Zero")
```

Instructions: Type a number when prompted (e.g., 10, -5, or 0).

Example Interaction & Output 1 (Positive):

```
Enter any number: 10
Positive
```

Example Interaction & Output 2 (Negative):

```
Enter any number: -5
Negative
```

Example Interaction & Output 3 (Zero):

```
Enter any number: 0
Zero
```

Part 7: Basic Loops (Repetitive Tasks)

Task 46: Counting Loop (For)

Use a `for` loop to print numbers from 1 to 5.

Type Python Code:

```
for i in range(1, 6):
   print(i)
```

Expected Output:

```
1
2
3
4
5
```

Task 47: Loop through List

Use a `for` loop to print each fruit in your fruit list, one per line.

Type Python Code:

```
my_fruits = ["apple", "banana", "cherry"]
for fruit in my_fruits:
    print(fruit)
```

Expected Output:

```
apple
banana
cherry
```

Task 48: While Loop

Use a `while` loop to print numbers from 1 to 3.

Type Python Code:

```
count = 1
while count <= 3:
    print(count)
    count = count + 1 # Important: ensures the loop eventually stops!
```

Expected Output:

```
1
2
3
```

Task 49: Count Down

Use a `while` loop to print numbers from 5 down to 1.

Type Python Code:

```
countdown = 5
while countdown >= 1:
    print(countdown)
    countdown = countdown - 1
```

Expected Output:

```
5
4
3
2
1
```

Task 50: Break a Loop

Use a `for` loop to print numbers from 1 to 10, but stop and print "Stopped at 6!" when the number reaches 6.

Type Python Code:

```
for num in range(1, 11):
    if num == 6:
        print("Stopped at 6!")
        break # This keyword immediately stops the loop
    print(num)
```

Expected Output:

```
1
2
3
4
5
Stopped at 6!
```

---

50 basic Python tasks with detailed code, instructions, and expected outputs. Take your time with each one. Experiment by changing the values or words. The best way to learn is by doing and seeing what happens! Good luck with your Python journey in PyCharm!