🐍 **Python Origins:** Created by Guido van Rossum in 1991, Python is renowned for its readability and flexibility. Named after the comedy troupe Monty Python, it aims to make coding fun and accessible.  
  
✨ **Magic of print():** In Python, the print() function serves as your magic spell for displaying text on the screen.  
  
📝 **String Enclosures:** Remember, strings must be enclosed in appropriate quotes.  
  
⚠️ **Case Sensitivity:** Python is [case-sensitive](https://hyperskill.org/learn/step/47352), so watch out for those details!  
  
🚫 **Indentation:** Beware of extra [indentation](https://hyperskill.org/learn/step/47352). This will lead to errors.

📞 **Call appropriately:** Calling functions by the wrong name is a crime in Python.

print(“james”)

print("Ever wonder why flamingos stand on one leg?\nIt's to conserve body heat!")

print('''My contact:

Email: your\_name@gmail.com

Phone: +xx-1234-5678''')

In our journey through Python's printing magic, we've discovered some cool tricks! Here's what we've been up to:

1. **Printing several lines:** The print() function is not just for printing one line – it's a versatile tool for creating [multi-line](https://hyperskill.org/learn/step/47555) output in our Python programs.
2. **Making spaces:** Using the print() function without anything inside the parentheses will leave an empty line.
3. **Being Quick:** Triple quotes or \n can be used to print multiple lines with a single print() function call.
4. **Using comments:** We can leave little notes in our code using #, kind of like leaving breadcrumbs for others to follow.

Always remember the golden rule: mix it up to keep it smooth!

print("Yes, I'm ready to learn Python.")

in Python, we've got **two** main types to play with: int for whole numbers and float for decimals.

print(type(100))

Alright, let's summarize what we've covered in this module — the ABCs of Python's basic [data types](https://hyperskill.org/learn/step/47516):

1. **Strings** (str): Think of these as the MVPs for handling text. They're your go-to when dealing with words and sentences. Plus, you can wrap them up in either single or double quotes for extra flexibility.
2. **Integers** (int): These are the plain ol' whole numbers, no decimal points involved. Whether it's -3, 0, or 42, they're your trusty companions for counting and beyond.
3. **Floats** (float): Now, these are the numbers with a bit of flair—they come with a decimal point. From 3.14 to -0.5, floats are your go-to for those precise measurements.

And hey, let's not forget about our buddy, the type() function. This little gem helps us figure out what type of data we're dealing with at any given moment.

message = "Good Morning!"

print(message)