🖨️ Print Function Overview

Brief Overview

This note covers **Python print function** and was created from the Lecture 4 : Learn the print() Function in 10 Minutes | Python for Beginners | #pythonprogramming YouTube video.
It covers basic syntax, multiple value printing, custom separators, string unpacking, and **escape sequences**.

Key Points

- Understanding the basics of print() and its case-sensitivity.
- Using commas to output multiple values and the default space separator.
- Customizing the separator with the sep parameter.
- Unpacking strings with * to print individual characters.

🖨️ Print Function Basics

Definition

> **print()** – a built-in function that writes the provided values to the standard output (screen).

Case Sensitivity

> Python is **case-sensitive**; the function must be written as print, not Print.

Simple Examples

- print(3) → 3
- print(4.5) → 4.5
- print('hello') → hello

```
print(3)
print(4.5)
print('hello')
```

➕ Printing Multiple Values

Using Commas

> Separate values with commas; Python prints them sequentially.

```
print('Jane is', 18)
print(3, 4, 5)
print('hello', 'world')
```

Default Separator

> By default, print inserts a single **space** between values (sep=' ').

🔧 Controlling the Separator (sep)

Syntax

```
print(value1, value2, ..., sep='your_separator')
```

Examples

```
print(1, 10, 2007, sep='/')   # 1/10/2007
print('I', 'U', 'N', sep='.') # I.U.N
```

Separator Table

Separator	Call Example	Output

space (default)	print(1, 2, 3)	1 2 3
/	print(1, 2, 3, sep='/')	1/2/3
.	print('I','U','N', sep='.')	I.U.N
\ (backslash)	print('a','b', sep='\\')	*problematic – may escape characters*

⭐ Unpacking Strings with *

> The * operator expands a string into its individual characters before they are sent to print.

```
print(*'UN')
```

Output

```
U N
```

↩️ Escape Sequences in Strings

Newline \n

> Inserts a line break, moving subsequent output to the next line.

```
print('Jane\nJohn')
```

Output

```
Jane
John
```

Tab \t

> Inserts a horizontal tab (default ≈ 4 spaces).

```
print('Jane\tJohn')
```

Output

```
Jane    John
```

Common Escape Sequences

Escape	Meaning	Example Result
\n	New line	Line1\nLine2 → two separate lines
\t	Tab (≈ 4 spaces)	A\tB → A B
