

What is a Subshell?

A **subshell** is a separate child process spawned by the parent shell. It allows executing commands in an isolated environment without affecting the parent shell.

How to Create a Subshell

Subshells are created using **parentheses** `()`.

Syntax:

```
(command1; command2; command3)
```

Example:

```
#!/bin/bash
echo "Parent Shell PID: $$"

(subshell_var="Hello from subshell"
echo "Subshell PID: $$"
echo $subshell_var)

# The variable is not available outside the subshell
echo "Subshell variable outside: $subshell_var"
```

Output:

```
Parent Shell PID: 12345
Subshell PID: 12346
Hello from subshell
Subshell variable outside:
```

The variable `subshell_var` is only available inside the subshell.

Differences Between a Subshell and a Parent Shell

Feature	Parent Shell	Subshell
Process ID (\$\$)	Remains the same	Different from the parent
Variable Persistence	Variables persist	Variables are lost after execution
Affects Parent?	Yes	No
Execution Scope	Global	Local

When to Use a Subshell

- To **isolate** commands and avoid modifying the parent environment.

- To **group** commands without affecting the current shell.
- To **run commands in parallel**.
- To **change directories** temporarily.

Example: Grouping Commands

```
#!/bin/bash
echo "Current directory: $(pwd)"

(cd /tmp; echo "Inside subshell: $(pwd)")

echo "Back to original: $(pwd)"
```