**Shell Scripting**

Shell scripting is a powerful tool for automating repetitive tasks and executing commands in Unix-like operating systems. Here are the basics:

* **Shell:** The shell is a command-line interpreter that takes commands from the keyboard and gives them to the operating system to perform. Common Unix shells include Bash (Bourne Again Shell), sh (Bourne Shell), csh (C Shell), and ksh (Korn Shell). Bash is the most common and widely used.
* **Script:** A shell script is a text file containing a sequence of shell commands that can be executed by the shell. Shell scripts typically have the .sh extension.
* **Shebang:** The first line of a shell script is called the shebang, which specifies the shell to be used to execute the script. For Bash scripts, the shebang line is #!/bin/bash.
* **Comments:** Comments in shell scripts start with the # symbol and are ignored by the shell. They are used to document the script's purpose and explain the code.
* **Variables:** Variables in shell scripts are defined without specifying a data type and are accessed using the $ prefix. For example:

NAME="John"

echo "Hello, $NAME!"

Command substitution: Command substitution allows you to capture the output of a command and store it in a variable. It is done using $(...) or backticks `...`. For example:

DATE=$(date)

echo "Today's date is $DATE"

Conditional statements: Conditional statements allow you to execute certain commands based on the result of a condition. Common constructs include if, elif, and else. For example:

if [ $1 -gt 10 ]; then

echo "Greater than 10"

else

echo "Less than or equal to 10"

fi

Loops: Loops allow you to execute a block of code repeatedly. Common loop constructs include for and while. For example:

for i in {1..5}; do

echo "Number: $i"

done

Functions: Functions allow you to group commands together and call them by name. They are defined using the function keyword or simply by naming the function followed by parentheses (). For example:

greet() {

echo "Hello, $1!"

}

greet "Alice"

File handling: Shell scripting provides various commands for file handling, such as creating, reading, writing, and deleting files. Common commands include touch, cat, grep, sed, awk, rm, mv, and cp.

These are just the basics of shell scripting, and there's much more you can do with it, including error handling, input/output redirection, and advanced scripting techniques.