# Welcome to the CoGrammar

## Cyber Security Tools: Bash Scripting

**The session will start shortly...**

**Questions? Drop them in the chat. We'll have dedicated moderators answering questions.**

CoGrammar

## Cyber Security Session Housekeeping

- The use of disrespectful language is prohibited in the questions, this is a supportive, learning environment for all - please engage accordingly. **(Fundamental British Values: Mutual Respect and Tolerance)**
- No question is daft or silly - **ask them!**
- There are **Q&A sessions** midway and at the end of the session, should you wish to ask any follow-up questions.
- If you have any questions outside of this lecture, or that are not answered during this lecture, please do submit these for upcoming Academic Sessions. You can submit these questions here: **Questions**

## Cyber Security Session Housekeeping cont.

- For all **non-academic questions**, please submit a query: www.hyperiondev.com/support
- We would love your **feedback** on lectures: Feedback on Lectures
- Find all the lecture **content** in you Lecture Backpack on GitHub.
- If you are hearing impaired, please kindly use your computer's function through Google chrome to enable captions.

# Safeguarding & Welfare

We are committed to all our students and staff feeling safe and happy; we want to make sure there is always someone you can turn to if you are worried about anything.

If you are feeling upset or unsafe, are worried about a friend, student or family member, or you feel like something isn't right, speak to our safeguarding team:

Ian Wyles
Designated Safeguarding Lead

Simone Botes

Nurhaan Snyman

Rafiq Manan

Ronald Munodawafa

Tevin Pitts

**Scan to report a safeguarding concern**

or email the Designated Safeguarding Lead:
Ian Wyles
safeguarding@hyperiondev.com

CoGrammar HyperionDev

## *Stay Safe Series:*

Mastering Online Safety One week at a Time

While the digital world can be a wonderful place to make education and learning accessible to all, it is unfortunately also a space where harmful threats like online radicalization, extremist propaganda, phishing scams, online blackmail and hackers can flourish.

As a component of this BootCamp the *Stay Safe Series* will guide you through essential measures in order to protect yourself & your community from online dangers, whether they target your privacy, personal information or even attempt to manipulate your beliefs.

## Security Tip

Close unused accounts

Delete old online accounts you no longer use. They may have weak passwords or poor data protection, making them easy targets for hackers. Clean up your digital footprint to protect your personal information.

CoGrammar

SKILLS FOR LIFE

SKILLS BOOTCAMPS

Department for Education

# Cyber Security Tools: Bash Scripting

November 2024

## Learning Objectives & Outcomes

**By the end of the lecture, everyone should be able to:**

- Explain the functionality and purpose of Bash scripting in Linux-based systems.
- Write a script that automates tasks like renaming files, managing directories, or processing user input.
- Define key components of a Bash script, such as the shebang, loops, and conditional statements.
- Create Bash scripts that combine multiple commands and logic structures to solve complex problems.

CoGrammar

## Learning Objectives & Outcomes

Can you think of any repetitive tasks or processes in your current work or personal life that could be simplified or automated?

CoGrammar

## Polls

Please have a look at the poll notification and select an option.

How familiar are you with Bash Scripting

A. I use it regularly for automation
B. I've tried it a few times
C. I'm aware of it but haven't used it
D. This is my first time exposure to bash scripting

CoGrammar

## Polls

Please have a look at the poll notification and select an option.

What is your preferred text editor for writing Bash Scripts?

A. Nano
B. Vi/Vim
C. Other
D. I haven't used a text editor for Bash Scripting yet

CoGrammar

## BASH SCRIPTING

- **Definition:** A series of commands executed line by line to automate tasks in the Bash shell.
- **Why Bash?**
  - Default shell in most linux distributions
  - Direct interaction with the system kernel
  - Faster and more versatile than GUI for many tasks

CoGrammar

## Setting Up For Bash Scripting

- **Shebang**
  - First line of a script to specify the interpreter
  - Example: #!/bin/bash
- **Make the script executable:**
  - **Command**: chmod +x script_name.sh
- **Run the script:**
  - **Command**: ./script_name.sh

CoGrammar

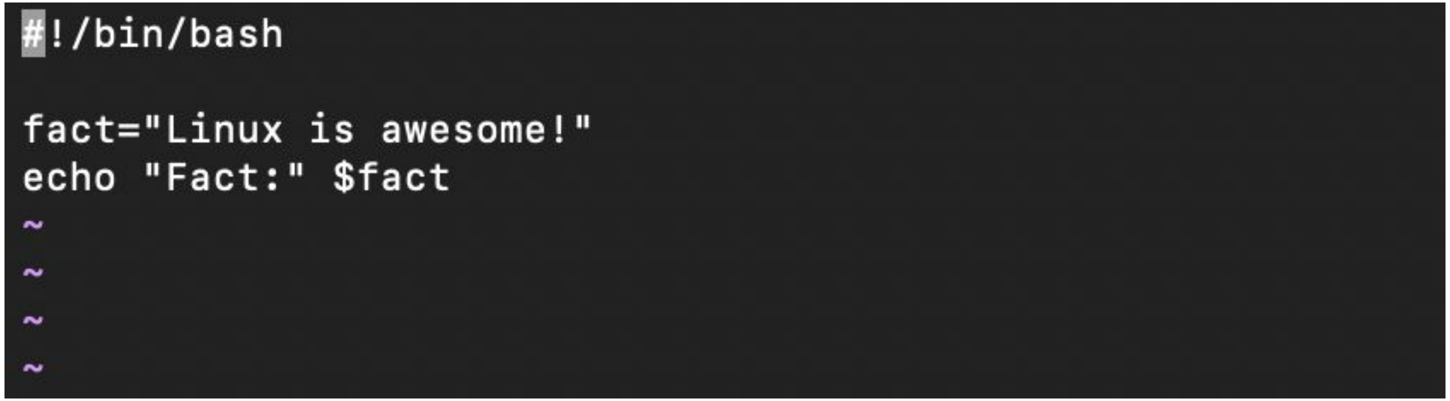
## Text Editors for Bash Scripting

- **Nano:**
  - Simple, user-friendly
  - Example Commands:
    - Save: Ctrl + O
    - Confirm: Enter
    - Exit: Ctrl + X
- **Vi/Vim:**
  - Powerful but steeper learning curve
  - Example Commands:
    - Insert mode: i
    - Save and exit: wq

CoGrammar

## Variables in Bash

- **Syntax**: variable_name=value
- **Example**:



```
#!/bin/bash

fact="Linux is awesome!"
echo "Fact:" $fact
~
~
~
~
```

CoGrammar

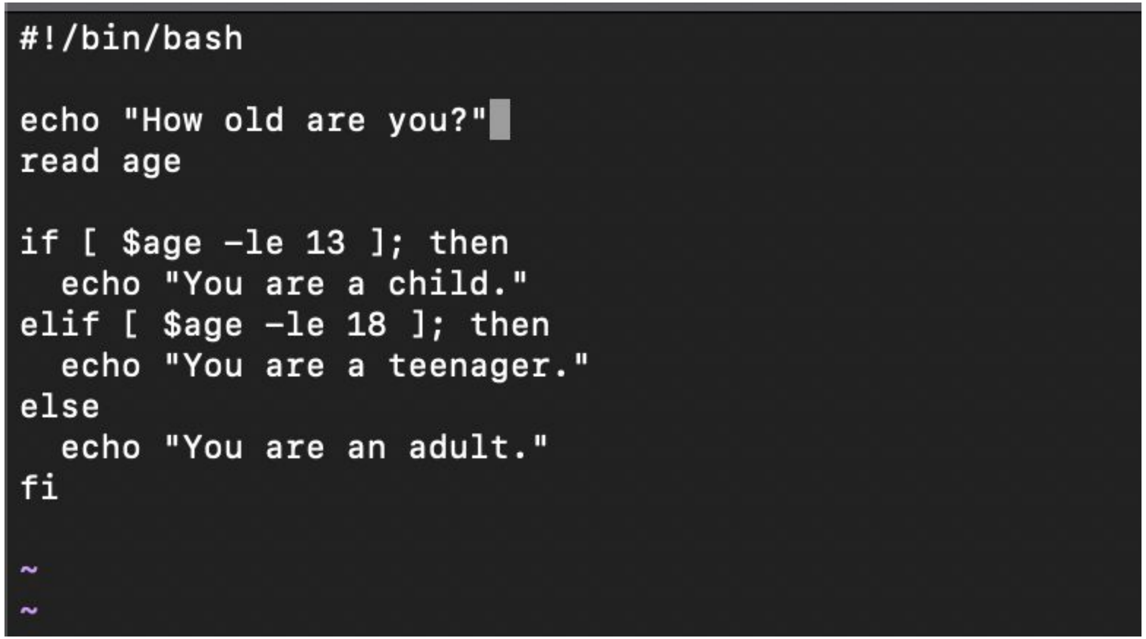
## Conditional Statements

- **Syntax:**

```
if [ condition ]; then
        statement
fi
```

CoGrammar

## Conditional Statements

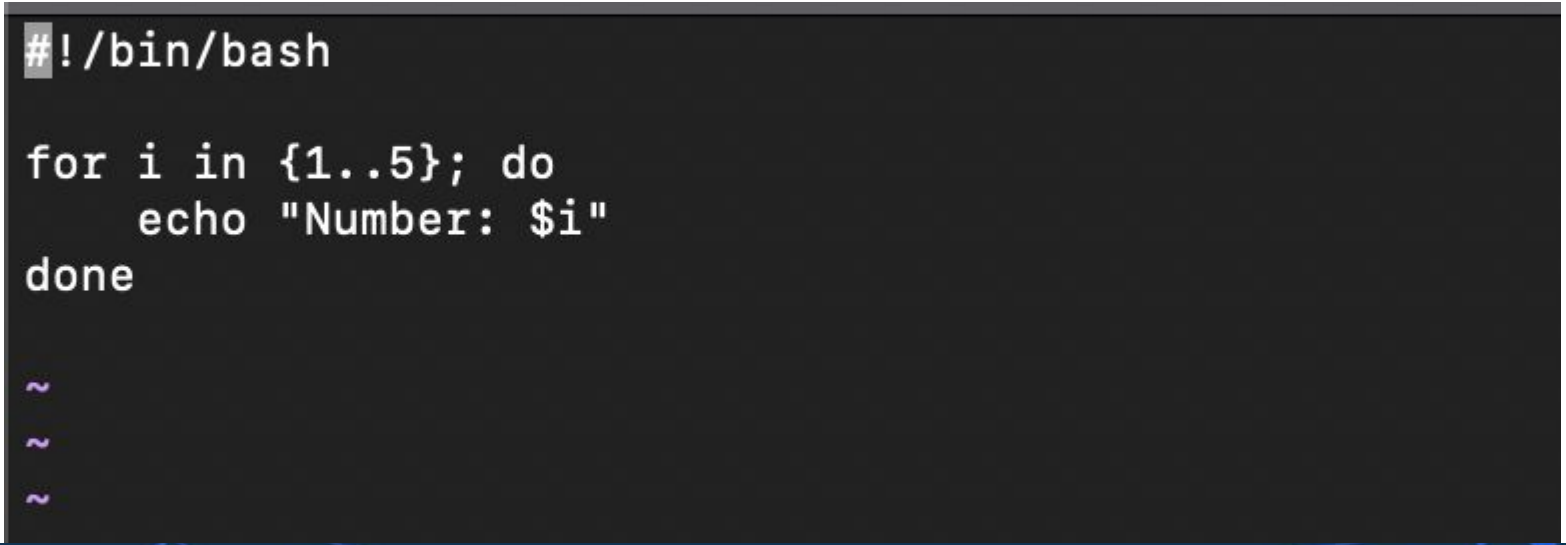


```
#!/bin/bash

echo "How old are you?"
read age

if [ $age -le 13 ]; then
  echo "You are a child."
elif [ $age -le 18 ]; then
  echo "You are a teenager."
else
  echo "You are an adult."
fi


~
~
```

CoGrammar

## Iterations in Bash

- Using a for loop:

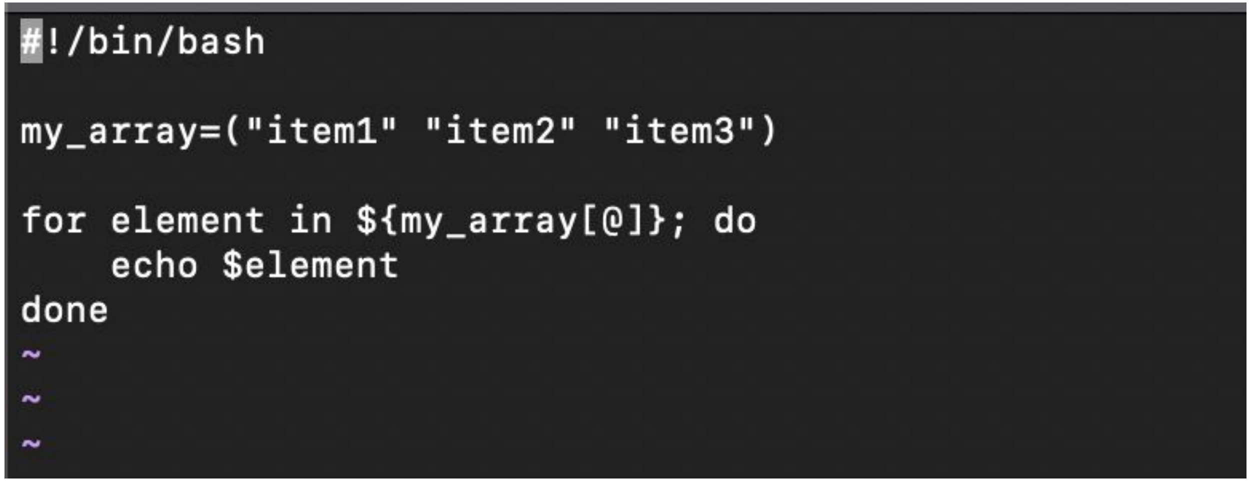


```
#!/bin/bash

for i in {1..5}; do
    echo "Number: $i"
done


~
~
~
```

CoGrammar

## Arrays in Bash

- Application: Process list of files, tasks, or other objects.



```
#!/bin/bash

my_array=("item1" "item2" "item3")

for element in ${my_array[@]}; do
    echo $element
done
~
~
~
```

CoGrammar

## Advantages of Bash Scripting

- **Automation**: Save time on repetitive tasks
- **Integration**: Combine commands into cohesive workflows
- **Portability**: Works across most unix-like systems
- **Ease of use:** Write scripts with basic text editors

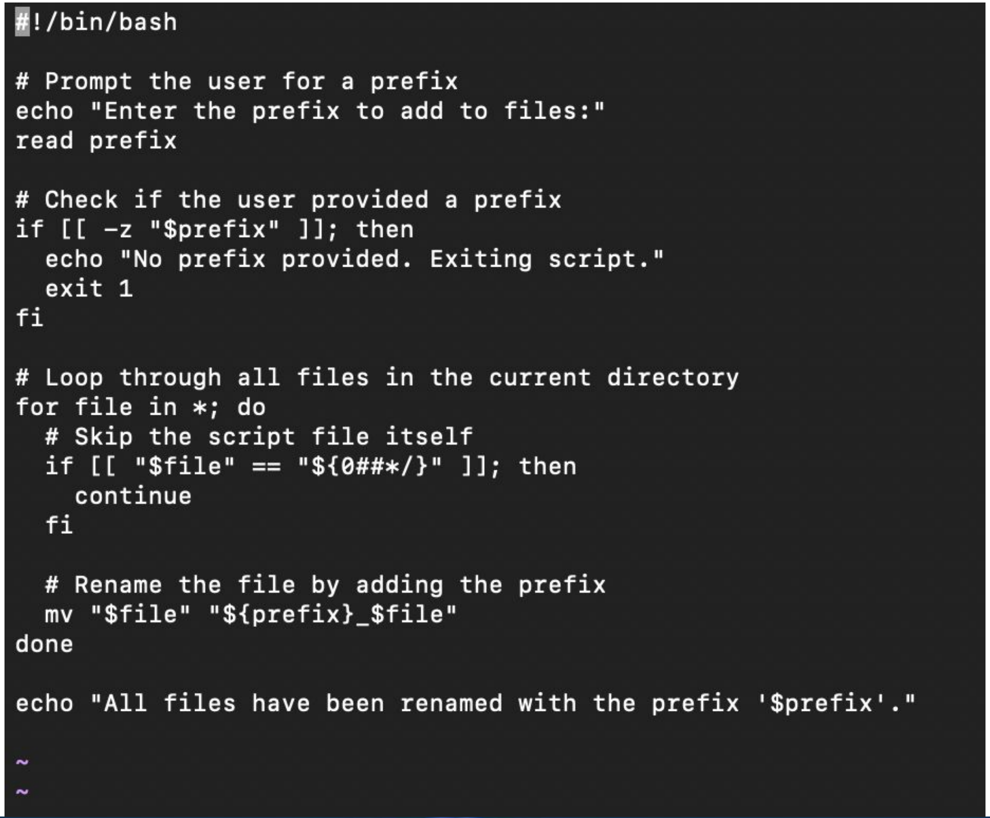
CoGrammar

## Practical Example: Batch File renaming

1. **Prompt for Input:** The script prompts the user to enter a prefix to prepend to filenames.
2. **Loop Through Files:** It uses a for loop to iterate over all files in the current directory (*).
3. **Rename Files:** Each file is renamed using the mv command, with the prefix added to its original name.
4. **Skip the Script:** The script ensures it doesn't rename itself by comparing filenames.

CoGrammar

## Practical Example: Batch File renaming



```
#!/bin/bash

# Prompt the user for a prefix
echo "Enter the prefix to add to files:"
read prefix

# Check if the user provided a prefix
if [[ -z "$prefix" ]]; then
  echo "No prefix provided. Exiting script."
  exit 1
fi

# Loop through all files in the current directory
for file in *; do
  # Skip the script file itself
  if [[ "$file" == "${0##*/}" ]]; then
    continue
  fi

  # Rename the file by adding the prefix
  mv "$file" "${prefix}_$file"
done

echo "All files have been renamed with the prefix '$prefix'."

~
~
```

CoGrammar

## Polls

Please have a look at the poll notification and select an option.

What is the purpose of the shebang (#!/bin/bash) in a Bash Script

A. To specify the script's file extension
B. To make the script executable
C. To indicate the interpreter to be used
D. To add a comment at the start of the script

CoGrammar

## Polls

Please have a look at the poll notification and select an option.

What operator is used in Bash scripting to represent “less than or equal to”?

A. -lt
B. -le
C. -eq
D. -ge

CoGrammar

## Questions and Answers

CoGrammar

# Thank you for attending

SKILLS FOR LIFE
SKILLS BOOTCAMPS

Department for Education

CoGrammar