**Knowledge Organiser:**

**Python Programming and Core Concepts**

**Introduction**

This knowledge organiser covers the essential Python programming concepts you need to know. These are important for your upcoming assessment.

**All activities are in GREEN, you are expected to write your answers in RED, evidencing your work with screenshots.**
**You can program online here: https://www.online-python.com/**

**Section 1: Introduction to Programming and Python**

Python is a high-level programming language known for its simplicity and readability. We use an environment called IDLE to create and run Python programs. The key terms we use when programming can be found below.

|  |  |  |
| --- | --- | --- |
|  **Key Term** | **Definition** | **Example** |
| Print | Display output to the user. | print("Hello World") |
| Assignment | Assigning a value to a variable | username = **"Bossman123"** |
| Variable | Storage location for data. | **username =** "Bossman123" |
| Input | Get information from the user. | age = input("Enter your age: ") |
| Concatenation | Joining strings together using “+” | print("Hello " + name) |
| F-String | Easier way to format strings with variables using “ | print(f"Hello {name}, you are {age} years old.") |
| Casting | Changing one data type to another. | int("10"), str(5.0) |
| Arithmetic | Performing mathematical calculations. | score = 10 + 5, remainder = 10 % 3 |
| Selection | Making decisions using if, elif, else. | if score >= 50: print("Pass") else: print("Fail") |
| Syntax Error | Mistakes in code rules. | primt("Hi") instead of print("Hi") — SyntaxError: name 'primt'... |

**DLE has 2 modes:**
1. Interactive mode – where you see your program results/the program being ran.



2. Script mode – where you create and write your program code.

**Section 2: Core Python Concepts**

**Printing to the screen:**

Use the print() function to display information.
*Example:*



**Triple Speech Marks:**

Use triple speech marks """ to create multi-line text.
*Example:*



**Variables:**

Variables are used to store information.
*Example:*



**Assignment Statements:**

The = sign is used to assign a value to a variable*Example:*

**

**Comments:**

Comments explain what code does. They start with a # symbol and are ignored by the computer.
*Example:*

**

# This prints Hello World

**Syntax Errors:**

Syntax errors happen when the rules of Python are not followed.

*Example:*

****.

**Activity (Complete this in RED)
Using the website:** [Online Python - IDE, Editor, Compiler, Interpreter](https://www.online-python.com/) complete the tasks below and evidence this with screenshots.

- Write 3 print statements that display different pieces of information about yourself.

|  |
| --- |
| **Place your screenshot of the code here**. (Screenshots can be taken by pressing prtscrn next to F12 and pasting, or using the snipping tool). |

- Use triple speech marks in one of your programs.

|  |
| --- |
| **Place your screenshot of the code here.** (Screenshots can be taken by pressing prtscrn next to F12 and pasting, or using the snipping tool). |

- Introduce and comment on 3 variables you create.

|  |
| --- |
| **Place your screenshot of the code here.** (Screenshots can be taken by pressing prtscrn next to F12 and pasting, or using the snipping tool). |

- Create 3 examples of syntax errors, and then correct them.

|  |
| --- |
| **Place your screenshot of the code here.** (Screenshots can be taken by pressing prtscrn next to F12 and pasting, or using the snipping tool). |

 **Section 3: String Handling - Concatenation and F-Strings**

**Concatenation**:
Concatenation uses + to join strings together.

*Example:*



In Python, you must convert numbers to strings when using concatenation because:

* The + operator only joins strings together when used for text.
* If you try to join a string ("You are ") with a number (14) directly, Python doesn't know how to "add" text and a number, so it gives an error:

This error is caused due to the lack of valid data types within the concatenation.

**F-Strings:**
F-Strings are newer, easier and allow the mixing of data types without having to convert them into strings.

*Example:*



To use an F-String, you start your string with the letter f, then place variables inside curly brackets {}:

*Python automatically converts numbers like age into strings when using F-strings — no need to use str().*

**Casting:**

Casting means changing a variable’s data type into another data type. This helps when we need to **calculate, compare**, or **display** values properly.

*Example:*



|  |  |  |
| --- | --- | --- |
| **Goal** | **Code Example** | **Explanation** |
| String ➔ Integer | age = int("14") | Turns string "14" into number 14 |
| String ➔ Float | pi\_value = float("3.14") | Turns string "3.14" into decimal 3.14 |
| Integer ➔ String | score = 100score = str(score) | Turns number 100 into string "100" |
| Float ➔ Integer | height = int(5.9) | Turns 5.9 into 5 (removes decimals) |
| Integer ➔ Float | points = 8points = float(points) | Turns 8 into 8.0 |

**Activity (Complete this in RED)
Using the website:** [Online Python - IDE, Editor, Compiler, Interpreter](https://www.online-python.com/) complete the tasks below and evidence this with screenshots.

- Create a program that prints out your name, age, and favourite subject using concatenation +

|  |
| --- |
| **Place your screenshot of the code here.** (Screenshots can be taken by pressing prtscrn next to F12 and pasting, or using the snipping tool). |

- Rewrite the program using an F-Strin { }

|  |
| --- |
| **Place your screenshot of the code here.** (Screenshots can be taken by pressing prtscrn next to F12 and pasting, or using the snipping tool). |

- Create three different varaibles (integer, string and float) then cast them to different values.

|  |
| --- |
| **Place your screenshot of the code here.** (Screenshots can be taken by pressing prtscrn next to F12 and pasting, or using the snipping tool). |

- Challenge: Ask the user for their favourite food and output a sentence using it inside an F-String.

|  |
| --- |
| **Place your screenshot of the code here.** (Screenshots can be taken by pressing prtscrn next to F12 and pasting, or using the snipping tool). |

**Section 4: Data Types**

A **data type** tells Python what kind of value is being stored in a variable.
Each type of data behaves differently — some can be used in maths, some store text, some store logic (True/False).

|  |  |  |  |
| --- | --- | --- | --- |
| **Data Type** | **Python Type Name** | **Example** | **What It Represents** |
| String | str | "hello" or 'abc' | Text (always inside quotation marks) |
| Integer | int | 42, -10, 0 | Whole numbers (used in maths) |
| Float | float | 3.14, 0.5, -2.0 | Decimal numbers (used for prices, weights, etc.) |
| Boolean | bool | True, False | Logic values — answers to yes/no questions |

Python needs to know the **data type** to:

* **Store and manage memory** correctly.
* Use the right **operations** (e.g. you can't add a string to an int without converting)
* Perform actions like loops, decisions, or calculations.

****If you don’t use the right data type, Python will give an error like:

**When and which data types do you use?**

|  |  |
| --- | --- |
| **When you want to...** | **Use this type:** |
| Store a name, message, or word | str |
| Do maths, counts, scoring using whole numbers | int |
| Do maths, counts, scoring using decimal numbers | float |
| Store a true/false answer (like “Is it raining?”) | bool |

**Real world examples:**

****

**Activity (Complete this in RED)
Using the website:** [Online Python - IDE, Editor, Compiler, Interpreter](https://www.online-python.com/) complete the tasks below and evidence this with screenshots.

- Task 1: Identify the data types used in the program code below:

|  |  |
| --- | --- |
| Variable Name: | Data Type: |
| a |  |
| b |  |
| c |  |
| D |  |
| e |  |

Task 2: Create a program that:

* Asks the user for their age
* Converts it to an integer
* Adds 5 years
* Prints their new age in years.

|  |
| --- |
| **Place your screenshot of the code here.** (Screenshots can be taken by pressing prtscrn next to F12 and pasting, or using the snipping tool). |

**Section 5: Arithmetic, BIDMAS, MOD and DIV**

**Arithmetic:
Arithmetic** in Python means using **mathematical operations** in your programs, like:

* Addition (+)
* Subtraction (-)
* Multiplication (\*)
* Division (/)

These operations are used to **calculate scores**, **health points**, **shopping bills**, **averages**, and much more. Python also follows BIDMAS rules.

|  |  |  |  |
| --- | --- | --- | --- |
| **Operator** | **Name** | **Example Code** | **Answer** |
| + | Addition | total = 5 + 3print(total) | 8 |
| - | Subtraction | lives = 10 – 2print(lives) | 8 |
| \* | Multiplication | points = 4 \* 6print(points) | 24 |
| / | Division (decimal result) | share = 8 / 2print(share) | 4.0 |
| // | DIV (whole number only) | whole = 17 // 3print(whole) | 5 |
| % | MOD (remainder after divide) | remainder = 17 % 3print(remainder) | 2 |

**What is DIV?**

* **DIV** means **floor division**.
* It divides two numbers but **removes the decimal** part.
* It **rounds down** to the nearest whole number.

*Example:*



**What is MOD?**

* **MOD** (modulus) gives the **remainder** after dividing two numbers.
* It tells you **what is left over** after division.

*Example:*

**

**BIDMAS:**

When you combine operations, Python follows the **BIDMAS** order:

* **B**rackets
* **I**ndices (powers)
* **D**ivision
* **M**ultiplication
* **A**ddition
* **S**ubtraction

*Example without brackets:

(multiplication happen first)*

 *Example with brackets:

(Brackets happen first)*

**Activity (Complete this in RED)
Using the website:** [Online Python - IDE, Editor, Compiler, Interpreter](https://www.online-python.com/) complete the tasks below and evidence this with screenshots.

- Create 3 different arithmetic calculations involving addition, subtraction, multiplication, and division.

|  |
| --- |
| **Place your screenshot of the code here.** (Screenshots can be taken by pressing prtscrn next to F12 and pasting, or using the snipping tool). |

- Program to practice MOD and DIV:

You bought a packet of sweets; it contains 88 sweets, and you want to share them equally between your 6 friends.
You want to know:
How many sweets **each person gets** (DIV)
How many sweets are **left over** (MOD)

|  |
| --- |
| **Place your screenshot of the code here.** (Screenshots can be taken by pressing prtscrn next to F12 and pasting, or using the snipping tool). |

- Challenge: You are responsible for **setting up a classroom** for a special event.
There are:

* **135 chairs** available in the school storeroom.
* You want to arrange them into **rows of 8 chairs per row**.

Your tasks:

1. Calculate **how many full rows** you can set up (use **DIV //**).
2. Calculate **how many chairs** will be **left over** without a row (use **MOD %**).
3. Display a clear message to show:
	* Number of full rows
	* Number of leftover chairs

|  |
| --- |
| **Place your screenshot of the code here.** (Screenshots can be taken by pressing prtscrn next to F12 and pasting, or using the snipping tool). |

**Section 6: Selection (IF, ELIF, ELSE)**

**Selection**:
Selection is a programming technique where your program **decides** based on a **condition using IF, ELIF and ELSE.** Depending on whether a condition is **True** or **False**, the program will choose different paths or different actions.

**Simple explanation:**
If something is true, do this. If not, do something else.

Without selection, a program would only run the same way every time.
Selection makes your programs **interactive**, **responsive**, and **smart** by choosing what happens based on inputs or calculations.

 **How selection works:**

Python checks conditions **one at a time**:

1. If the **if** condition is **True**, it runs the indented code underneath.
2. If **if** is False, it checks the **elif** (else if) conditions, one by one.
3. If no **if** or **elif** condition is True, the **else** code runs automatically.

*Example:*



**Explanation:**

* If score > 90, print "Grade 9"
* If not, but score > 70, print "Grade 7"
* If neither, print "Grade 5"

|  |  |
| --- | --- |
| **Rule** | **Explanation** |
| Always use a **colon:** | After if, elif, or else, you must use a colon. |
| **Indent the next line** | After the colon, the code inside the decision is indented. Failure to use indents results in a syntax error. |
| Indent using **4 spaces** or **Tab** | Python needs indents to group code together. |
| Use **comparison operators** like ==, >, <, >=, <=, != | For checking conditions. |

**What happens if you forget to indent?**

**Activity (Complete this in RED)
Using the website:** [Online Python - IDE, Editor, Compiler, Interpreter](https://www.online-python.com/) complete the tasks below and evidence this with screenshots.

Task 1: Create a program to evaluate your knowledge of IF, ELIF and ELSE statements. Remember to follow the rules listed above, using colons and indents. The program should:

* Asks the user for their **score** out of 100.
* If they score **90 or above**, print "**Excellent**".
* If they score **between 70 and 89**, print "**Good job**".
* Otherwise, print "**Keep trying**”.

|  |
| --- |
| **Place your screenshot of the code here.** (Screenshots can be taken by pressing prtscrn next to F12 and pasting, or using the snipping tool). |

*Hint:*


Task 2: Create a program to evaluate your knowledge of IF, ELIF and ELSE statements. Remember to follow the rules listed above, using colons and indents. The program should:

* Asks for the user's **age**.
* If age is **18 or over**, print "**You can vote**."
* If age is **16 or 17**, print "**You can learn to drive**."
* Otherwise, print "**You are still in school!**"

|  |
| --- |
| **Place your screenshot of the code here.** (Screenshots can be taken by pressing prtscrn next to F12 and pasting, or using the snipping tool). |

Challenge Task: Create a program to evaluate your knowledge of IF, ELIF and ELSE statements. Remember to follow the rules listed above, using colons and indents. The program should:

* Asks the user "**Is it raining?**" (answer yes or no)
* If yes, ask "**Is it windy?**" (answer yes or no)
* If raining and windy, print "**Too windy for an umbrella**."
* If raining but **not windy**, print "**Take an umbrella**."
* If not raining, print "**Enjoy your day!**"

|  |
| --- |
| **Place your screenshot of the code here.** (Screenshots can be taken by pressing prtscrn next to F12 and pasting, or using the snipping tool). |

**Section 7: Assignment vs Comparison**

**Assignment vs Comparison:**

**Assignment** means **giving a value to a variable**, it does this using =.

* score is assigned the value 10.
* name is assigned the value "Alice".

**Comparison** means **checking if two things are equal**, it does this using ==

****

* We are **comparing** whether score is equal to 10.
* Comparison is always used inside **if**, **elif**, and **while** statements.

**Comparison operators:**

These operators are used inside **if, elif, while, and assert`** statements to make decisions in your program.

|  |  |  |  |
| --- | --- | --- | --- |
| **Operator** | **Name** | **Example** | **What it Checks** |
| == | Equal to | score == 10 | Is score **exactly equal** to 10? |
| != | Not equal to | score != 0 | Is score **not equal** to 0? |
| > | Greater than | score > 50 | Is score **more than** 50? |
| < | Less than | score < 30 | Is score **less than** 30? |
| >= | Greater than or equal to | score >= 70 | Is score **70 or more**? |
| <= | Less than or equal to | score <= 100 | Is score **100 or less**? |

**Activity (Complete this in RED)
Using the website:** [Online Python - IDE, Editor, Compiler, Interpreter](https://www.online-python.com/) complete the tasks below and evidence this with screenshots.

Task 1: Correct the program below by changing the operator and used to give a true result. The first one has been done for you.



|  |
| --- |
| **Place your screenshot of the code here.** (Screenshots can be taken by pressing prtscrn next to F12 and pasting, or using the snipping tool). |

Task 2: Ask the user to enter a number and depending on the number deliver a different response using operators like in the example below.

* If the number is higher than 10, print out “your number is higher then 10!”.
* If the number is higher than 50, print out “your number is higher then 50!”
* Finally, if the number is lower than 0, print out “your number is below 0”.



|  |
| --- |
| **Place your screenshot of the code here.** (Screenshots can be taken by pressing prtscrn next to F12 and pasting, or using the snipping tool). |

**Section 8: Loops – FOR and WHILE (Extension)**

A **loop** is used when you want your code to **repeat** something — either a **set number of times** or **until a condition is met**.

Python has **two main types of loops**:

|  |  |
| --- | --- |
| **Loop Type** | **Used When** |
| for loop | You know **exactly how many times** to repeat. |
| while loop | You want to **repeat until a condition is False**. |

 **For Loops:**For loops are when you want to repeat something a set number of times (e.g. 5 times).

*Example:*



* **The letter i within the code is just a variable name.**
* It stands for **index** or **iteration**, but we can call this variable anything.
* It helps the loop **keep count** and **changes each time** the loop runs.
* This lets Python know **how many times to repeat** the code.
* range(5) means it loops 5 times from 0 to 4.

*Example 2:*



**While Loops:**Use while loops when the number of repeats **depends on a condition**.

*Example:*



The code starts with 5 bullets
Each time the loop runs:

* It prints how many bullets are left.
* Takes 1 away.
* When ammo reaches 0, it stops and prints a message.

*Example 2:*



* The code starts with a correct password.
* The user is then asked to enter a password which is stored as a second variable.
* While the second variable does not match the correct password, the user will be asked to retry and input the password again.
* The while loop will run this code forever UNTIL it matches the correct password.

**Activity (Complete this in RED)
Using the website:** [Online Python - IDE, Editor, Compiler, Interpreter](https://www.online-python.com/) complete the tasks below and evidence this with screenshots.

Task 1: Create a for loop that counts down from 5 to 1 and prints:



*Hint: Use range() to count down (e.g. range(5, 0, -1))*

|  |
| --- |
| **Place your screenshot of the code here.** (Screenshots can be taken by pressing prtscrn next to F12 and pasting, or using the snipping tool). |

*T*ask 2: Create a for loop that prints out how much ammo the user has left when its above 0. If the Ammo reaches 0, the program should print out “Out of Ammo! Time to reload!”

*Hint:*





|  |
| --- |
| Place your screenshot of the code here. (Screenshots can be taken by pressing prtscrn next to F12 and pasting, or using the snipping tool). |

**Section 9: Photoshop and Non-Destructive Editing**

**What is Photoshop?**

**Photoshop** is a powerful image-editing program used in industries like graphic design, marketing, web development, and digital art. It allows users to manipulate, enhance, and combine images using tools like **layers, masks, adjustments, and brushes**.

|  |  |
| --- | --- |
| **Tool / Concept** | **What it Does / Why It's Used** |
| **Layer** | Lets you work on one part of the image without changing the rest — like stacking transparent sheets |
| **Adjustment Layer** | Applies effects (like brightness, contrast, colour) **without changing** the original image |
| **Mask** | Controls **where** an adjustment appears: **white shows**, **black hides** |
| **Brush Tool** | Paints or removes parts of a mask or layer — size, opacity, and hardness can be adjusted |
| **Lasso Tool** | Makes **freehand selections** around parts of the image |
| **Clone Tool** | Copies pixels from one part of the image to another — great for fixing or duplicating |
| **Spot Healing Brush** | Automatically removes **blemishes, marks, or scratches** by blending with nearby pixels |
| **Hue/Saturation** | Adjusts colour tone or intensity — useful for **colour correction** or artistic changes |
| **Opacity** | Makes a layer or brush **see-through** — 100% is solid, 0% is invisible |
| **Liquify Tool** | Warps, bends, or reshapes parts of an image — often used to subtly adjust facial or object shapes |

**Common Keyboard Shortcuts:**

|  |  |
| --- | --- |
| **Action** | **Shortcut** |
| Undo | CTRL + Z |
| Deselect (selection) | CTRL + D |
| Stop Lasso Tool | ESC |
| Resize brush | [ ] (brackets) |
| Zoom In/Out | ALT + Scroll |

**Non-Destructive editing:**

**Non-destructive editing** means editing **without damaging** the original image.
This is done by using:

* Layers (instead of editing directly)
* Adjustment layers (effects applied on top)
* Masks (to control where the changes happen)

**Photoshop UI and Icons:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **#** | **Tool Name** | **Shortcut** | **What It Does** |
| 1 | **Move Tool** | V | Moves layers, images, or selections |
| 2 | **Marquee Tool** | M | Makes rectangular or elliptical selections |
| 3 | **Lasso Tool** | L | Freehand selection tool for tracing around objects |
| 4 | **Quick Selection Tool / Magic Wand** | W | Selects areas based on colour or tone similarity |
| 5 | **Crop Tool** | C | Trims or resizes the image canvas |
| 6 | **Slice Tool** | K | Used for dividing an image for web use (rarely used in schools) |
| 7 | **Eyedropper Tool** | I | Picks a colour from the image to use as your foreground colour |
| 8 | **Spot Healing Brush Tool** | J | Removes blemishes and small marks by blending with nearby pixels |
| 9 | **Brush Tool** | B | Paints on the canvas or mask with customisable size and softness |
| 10 | **Clone Stamp Tool** | S | Copies pixels from one area to another — used for duplicating or retouching |
| 11 | **History Brush Tool** | Y | Paints a previous state of the image onto the current version |
| 12 | **Eraser Tool** | E | Erases parts of a layer or mask |
| 13 | **Gradient or Paint Bucket Tool** | G | Fills areas with a solid colour or gradient |
| 14 | **Blur Tool** | No default | Softens hard edges or details |
| 15 | **Dodge Tool** | O | Lightens parts of the image — used for highlights |
| 16 | **Pen Tool** | P | Creates paths or shapes — used for precise selections or design work |
| 17 | **Type Tool** | T | Adds text to your image |
| 18 | **Path/Direct Selection Tool** | A | Selects and moves anchor points on shapes or paths |
| 19 | **Line Tool** (or Shape Tool) | U | Draws lines or vector shapes |
| 20 | **Hand Tool** | H | Moves the canvas around (doesn’t move the image) |
| 21 | **Zoom Tool** | Z | Zooms in or out on the canvas |
| 22 | **Foreground/Background Colour** | D (reset) | Shows and switches between your active colours |

**Activity (Complete this in RED)
Using the website:** [Online Python - IDE, Editor, Compiler, Interpreter](https://www.online-python.com/) complete the tasks below and evidence this with screenshots.

Task 1: You are editing a car advert. The client wants the red car to appear blue but isn’t sure exactly what shade they want yet. They’d like you to show a version that they can **adjust later without starting over**.

What tool(s) and process would you use to **change the colour** of the car **without damaging the original image**?

Hint: Think about non-destructive tools like adjustment layers and selection tools.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Task 2: You’ve been asked to retouch a close-up headshot for a school website. The subject has a few small skin blemishes and would like them removed, but they want the image to look **natural and untouched** — not over-edited.

Which tool(s) would you use to remove the blemishes while keeping the skin looking natural?
Explain why this is a better choice than simply painting over the area.

*Hint: Think about tools that blend with the surrounding area automatically.*

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_