01 Zero & One

Unit 13: AI Programming

Programming & AI, 1st secondary (Introduction to ICT)



Engineer Rahim

13-1 About Generative AI

- **What is generative AI?**: **Generative AI:** performs intellectual tasks on behalf of people, so using it well makes your **workflow** easier and faster.
- **How does it write?**: It has learned from a huge amount of information
- **Hallucination**: Because of this way of working, it is **certain** to say wrong things some of the time, even when it sounds confident
- **Tools and prompts**
  - Well-known tools: **ChatGPT**, **Google Gemini** and **Microsoft Copilot** (the one we use in this unit).
  - You talk to it in **everyday language**: "Write a story about...", then another prompt in the **same chat** changes the same text: "Add a character to the story...".
  - It can make you a **quiz**: you choose the number of questions, their type (true/false, multiple choice, open-ended), that it **waits for your answer**, and that it shows the **score** at the end.
  - Try the quiz on someone else, and change the topic or the number of questions with a new prompt.

Generative AI	AI that performs intellectual tasks and generates new text, code or images
Prompt	The instruction a user writes to the AI
Hallucination	When the AI gives wrong information as if it were true
Workflow	The sequence of steps you follow to get work done
Most likely next word	The word with the highest probability of coming next; the basis of how AI builds text
Microsoft Copilot	Microsoft's generative AI tool used in this unit
Quiz	A short set of questions the AI can generate for practice

13-2 HTML, CSS

- **HTML and CSS**: **HTML (HyperText Markup Language):** the language that displays web pages in the browser
- **Tags and the page layout**
  - Each element is written between **tags**, such as `<h1>…</h1>` for a heading and `<p>…</p>` for a paragraph.
  - `<!DOCTYPE html>` at the top tells the browser this is HTML.
  - `<head>`: information about the page that is not shown inside it, such as `<meta charset="UTF-8">` and `<title>` (the page name shown on the **browser tab**).
  - `<body>`: the content that actually appears on the page.
- **Linking CSS to the page**: In `<head>` we write `<link rel="stylesheet" href="style.css">` so the page reads its design from the file `style.css`.
- **The class**: You write `class="name"` in the tag, and in CSS you write `.name { color: blue

HTML	HyperText Markup Language; builds the structure of a web page
CSS	Cascading Style Sheets; defines the design of a web page

Tag	A marker such as <p> that defines an element in HTML
<head>	The part of the page holding information that is not displayed in it
<body>	The part of the page holding the content shown in the browser
<title>	The page name shown on the browser tab
<link>	A tag that connects the page to an external CSS file
class	An attribute that names elements so CSS can style them with .name
WebDen	A website for writing and running HTML, CSS and JavaScript code

13-3 HTML, JS

- **What does JavaScript do?**: **JavaScript:** is a programming language that runs **inside the browser**
- **id, getElementById and textContent**: You give a tag a **unique identifier** `id="name"`, and in JavaScript you get it with `document.getElementById("name")`.
- **Working with a button**
  - `const btn = document.getElementById("btn");` saves the button in a variable.
  - `btn.onclick = () => { … };` the code inside the braces `{ }` runs **only when the button is clicked**.
  - `text.style.fontSize = 50 + "px";` makes the font bigger: `50 + "px"` becomes the text `"50px"`.
  - `p.style.color = "red";` changes the text color.
- **Ideas to try**
  - Change the color and the size together with the same click.
  - Make a **function** that joins the text of two elements when you press the button.
  - Make the color **alternate** between red and blue on each click using `if … else`.

JavaScript	A programming language that runs in the browser and makes pages interactive
id	A unique name for one element so JavaScript can reach it
getElementById	A function that gets an element from the page by its id
textContent	A property that reads or changes the text inside a tag
onclick	The code that runs when the element is clicked
style	A property used to change an element's look from JavaScript
Interactive	Changing and responding to the user's actions
Function	A named block of code that runs when it is called

13-4 Creating Static Web Pages

- **Programming is a strength**: **Programming:** is one of generative AI's biggest strengths: you give it instructions and it writes the program for you.
- **What is a static page?**: A **static web page** is made with **HTML and CSS only**
- **Working steps**
  - **1) Generate:** open Copilot and type a prompt like "Create a simple self-introduction page using HTML and CSS".
  - **2) Run:** copy the code, paste it into **WebDen** and press **Run**
  - **3) Modify:** change the HTML and CSS (the name, text color, font size, background color) so the page becomes yours.

    - **4) Understand:** ask it "Explain this code."
    - **5) Organize:** rename the chat (the three dots "..." → Rename) so you can return to it easily later.
- **Exercise**: Try pages for different purposes: a news page, a ranking of popular sports, restaurants in your area, an introduction to your school, or famous tourist spots in Egypt.

Static web page	A page built with HTML and CSS whose content does not change with user actions
Generated code	Code written by AI; it may contain errors and must be tested
Run	The WebDen button that executes the code and shows the result
Rename	Giving a chat a new name so it is easy to find later
Customize	Editing the code so the page fits your own needs
background-color	A CSS property that sets the background color

13-5 Creating Dynamic Web Pages

- **The dynamic page**: A **dynamic (interactive) web page** changes and responds to the user, and this interaction comes from **JavaScript**.
- **Example: changing the background color**
    - A prompt like: "Create a self-introduction page using HTML, CSS and JavaScript, and make the background color change when a button is pressed."
    - `document.body.style.backgroundColor = "yellow";` changes the background of the whole page.
    - The colors can be stored in an **array** such as `["red", "green", "blue"]`
    - `addEventListener("click", …)` runs the code when the element is clicked.
- **Example: a news page**: A prompt like: "Write a simple news page where the main text switches when a headline is clicked." Then add more articles or make the background color change with the article shown.
- **Learn from the code**: Did you understand the code it wrote? **Write down the unclear parts** and look them up or ask about them

Dynamic web page	A page that changes and responds to the user, using JavaScript
addEventListener	A function that links an event, such as a click, to code that runs when it happens
Array	An ordered list of values, such as a list of colors
backgroundColor	The JavaScript style property that changes an element's background color
Event	Something that happens, such as the user clicking a button
Headline	The title of a news article

13-6 Presenting Your Website

- **What will you present?**: In this lesson you show the site you made last class to **your group members**
- **HTML**
    - `<link>` loads a CSS file, and `<script>` loads a JavaScript file into the page.
    - If you give a `<button>` an `id`, you can control it from JavaScript.
- **CSS**
    - The `body` selector sets the background color of the whole page.

- The `button` selector sets the color of buttons and their animations.

- **JavaScript**
  - `document.getElementById` selects a specific element in the HTML, such as a button or a text.
  - `addEventListener` is how you write what happens when the button is clicked.

<script>	A tag that loads a JavaScript file into an HTML page
Selector	The part of a CSS rule that chooses which elements to style
body selector	A CSS selector that styles the whole page, such as its background
Animation	Movement or a gradual change in how an element looks
Presentation	Explaining your work to your classmates

13-7 Developing a Quiz App

- **First version**
  - A prompt like: "Create a simple multiple-choice quiz app with only one question, with separate HTML, CSS and JavaScript code."
  - Paste each code into **its own tab** in WebDen and press Run (the code differs every time).
  - Change the question text or the answer, ask "Explain this code.", and rename the chat to "Developing a Quiz App".
- **3 questions**: Tell it "Make it 3 questions." The questions are usually stored in an **array** with one **object** per question: the question text, the choices, and the correct answer
- **Checking the answer and the score**: With `if (picked === answer) { … } else { … }` it decides whether to write "Correct" or "Wrong"
- **Improve the app**
  - **Increase the number of choices:** (add an item to the choices array).
  - **Show the percentage of correct answers:** at the end.
  - **Randomize the order of the questions:** each time.

Quiz app	A web app that shows questions and checks the answers
Object	A group of named values, such as question, choices and answer
length	A property that gives the number of items in an array
if...else	A statement that runs one part if a condition is true and another part if not
Score	The number or percentage of correct answers
Random order	Showing the questions in a different order each time

13-8 Quiz App Review

- **Summarize in 4 points**: **The quiz theme:** (space, animals...), and what you learned in **HTML**, **CSS** and **JavaScript**.
- **HTML and CSS**
  - `<title>` sets the app name shown on the tab.
  - Giving an element a `class` lets you refer to it from CSS.
  - The `body` selector sets the background color and the font of the whole page.
  - `.choice` (a class selector) sets the size and color of the choice buttons.
  - `background: #f5f5f5;` is a very light gray background (a hex color code).

- **JavaScript**
  - The many choices are stored in an **array** named `choices`.
  - The text color changes for a right or wrong answer with an `if...else` statement.
  - `<p id="result"></p>` is an empty paragraph where JavaScript writes the result.
- **Sort and ask**: Split your summary into "things I knew" and "things I didn't know", and ask the AI about the second group, for example: "What does this code do?" or "What does `background: #f5f5f5;` mean?".

Class selector	A CSS selector such as .choice that styles elements with that class
Hex color	A color written as a hexadecimal code such as #f5f5f5
choices array	An array that stores the options of a question
Result paragraph	An empty element where JavaScript writes the result
Review	Summarizing and sorting what you learned

13-9 Developing a Block Breaker Game 1

- **Generating the game**
  - Open Copilot and type: "Create a simple block breaker game, with separate HTML, CSS and JavaScript code."
  - Paste the code into its tabs in WebDen and run it (the result differs from one student to another).
  - Ask "Explain this code." and ask yourself: how much did I understand?
  - Rename the chat to "Block Breaker Development" so you can continue in it next lesson
- **Edit by hand**: This time **without AI**: think about the program and edit it yourself: **make the ball bigger**, **change the color of the blocks**, **increase the number of blocks**.
- **Difficulties and strategies**
  - **Difficulties:** not knowing which value makes the ball bigger
  - **Strategies:** guess the meaning from **variable names** and look for the related parts

Block breaker game	A game where a ball bounces off a paddle to break blocks
Blocks	The bricks the ball breaks
Paddle	The bar the player moves to bounce the ball back
Variable name	The name of a variable; it helps you guess what a part of the code does
ballRadius	A common variable holding the ball's radius (its size)
fillStyle	A canvas property that sets the drawing color

13-10 Developing a Block Breaker Game 2

- **Continue in the same chat**: Open the "Block Breaker Development" chat from last lesson so the AI remembers **the earlier code**
- **Features to add**
  - **Ball speed:** multiply `dx` and `dy` by a number (for example 1.5 = one and a half times faster).
  - **Score:** a variable that goes up each time a block breaks.
  - **Lives:** a variable that goes down when the ball falls
  - **Block durability:** a block needs several hits before it breaks.
  - You can let the AI write the code, or edit it by hand.

- **Prepare for the presentation**
  - What game did you make?
  - Creative points and difficulties.
  - What can AI do, and what can it not do?

Lives	The number of times the ball may fall before the game ends
Durability	How many hits a block needs before it breaks
Angle	The direction the ball moves after a hit
Ball speed	How fast the ball moves, set by dx and dy
Context	What the AI remembers from earlier code and messages in a chat

13-11 Block Breaker Summary

- **1) The game**: **The game title:** and **its rules**: number of lives, ranking, whether there are items...
- **2) Development stories**
  - **Creative points:** for example, the ball speed changes each time it hits the paddle, or the score text changes color when you beat the high score.
  - **Difficulties:** for example, adding more blocks made the game too hard, or the program stopped halfway and I had to rebuild it from the start.
- **3) Developing with AI**: What it **can** do (write code fast, explain, help summarize), and what it **cannot** do (feel whether the game is fun, guarantee the code has no bugs, present for you)
- **Put it into a presentation**: Summarize all of this in **4 to 5 slides**

Game rules	How the game is played, such as lives, ranking and items
Development stories	The creative points and difficulties met while building the game
High score	The best score reached so far
Items	Objects in the game that give the player a bonus
Slides	The pages of a presentation

13-12 Block Breaker Development Presentation

- **What will you present?**
  - **1) The game:** the title and the rules.
  - **2) Development stories:** creative points and the difficulties you faced.
  - **3) Collaboration with AI:** what it can do, and what it cannot do.
- **You and the AI**: The AI sped up writing the code, but **you** decided what to build, **tested** and fixed it, and **judged** whether the game was fun
- **The presentation**: Present your summary to **your group members**, and listen to their presentations and learn from their ideas.

Collaboration with AI	Working together with AI, sharing the work between you and it
Creative points	The ideas and strategies you added yourself
Difficulties	The problems you faced during development
Group members	The classmates you present to

Study the whole course free

Every lesson, questions with answers, a ministry-style bank and a timed mock exam

zerooneacademy.com

Engineer Rahim