ملخص الوحدة 12: البرمجة

البرمجة والذكاء الاصطناعي، أولى ثانوي

مهندس رحيم

12-1 الخوارزميات

- **الخوارزمية وطرق رسمها: الخوارزمية (Algorithm):** = طريقة أو خطوات لحل مشكلة معينة.
- **رموز المخطط الانسيابي**
  - **Terminal:** البداية / النهاية.
  - **Display:** عرض على الشاشة.
  - **Data:** إدخال وإخراج البيانات.
  - **Process:** العمليات والحسابات.
  - **Conditional branch:** تفرّع حسب شرط (Yes / No).
  - **Repeat:** بداية التكرار ونهايته.
- **رموز مخطط النشاط**
  - **Start:** و **End:** بداية ونهاية النشاط.
  - **Control:** خطوة أو إجراء بيتعمل.
  - **Transition:** سهم الانتقال من خطوة للتانية.
  - **Conditional branch:** تفرّع حسب شرط.
  - **Parallel process:** خطوات بتحصل في نفس الوقت.
  - **Send:** إرسال إشارة أو رسالة لطرف تاني.
- **هياكل التحكم التلاتة**
  - **الهيكل التسلسلي (Sequential):** الخطوات بتتنفذ بالترتيب واحدة ورا التانية.
  - **الهيكل التكراري (Repeating):** بيكرر العملية طول ما الشرط متحقق.
  - **الهيكل المتفرع (Branching):** بيختار عملية حسب الشرط
  - أي خوارزمية ممكن تتكتب بالتلاتة دول مع الإدخال والإخراج.
- **لغات البرمجة**
  - **Python:** للذكاء الاصطناعي والإحصاء
  - **JavaScript:** بتشتغل جوه متصفح الويب
  - **Scratch:** لغة مرئية للتعليم

Algorithm	الخوارزمية: طريقة أو إجراء لحل مشكلة معينة
Flowchart	مخطط انسيابي: بيوضح تدفق عملية واحدة
Activity diagram	مخطط النشاط: مناسب للعمليات المتوازية
Sequential structure	الهيكل التسلسلي: الخطوات بالترتيب
Repeating structure	الهيكل التكراري: تكرار طول ما الشرط متحقق
Branching structure	الهيكل المتفرع: اختيار عملية حسب الشرط
Programming language	لغة البرمجة: لغة للتعبير عن الخوارزميات للكمبيوتر
Machine language	لغة الآلة: أوامر من أصفار ووحايد يفهمها الكمبيوتر
Python	لغة للذكاء الاصطناعي والإحصاء بأقل كود
Scratch	لغة برمجة مرئية تعليمية بالبلوكات

12-2 أساسيات البرمجة [1]

- **الدالة print**: `print()` بتعرض النص أو القيمة اللي جوه القوسين
- **المتغير ومعامل التخصيص**: **المتغير (Variable):** زي صندوق بنخزّن فيه بيانات ونرجع ناخدها
- **المعاملات الحسابية**
  - `a + b` جمع
  - `a / b` قسمة
  - `a // b` **خارج القسمة** الصحيح: `7 // 2 = 3`.
  - `a % b` **باقي القسمة**: `7 % 2 = 1`.
  - `a ** b` الأُس (**a أُس b**): `3 ** 2 = 8`.
- **الهيكل التسلسلي**: البرنامج بيتنفذ **سطر سطر من فوق لتحت**

print()	دالة بتعرض النص أو القيمة اللي جوه القوسين
Variable	المتغير: صندوق لتخزين البيانات واسترجاعها
Assignment operator	معامل التخصيص (=): بيحط قيمة اليمين في متغير الشمال
Arithmetic operator	المعامل الحسابي: رموز الحساب زي + و -
String	سلسلة الأحرف (نص) بين علامتي تنصيص
Quotient (//)	خارج القسمة الصحيح
Remainder (%)	باقي القسمة
Power (**)	الأُس: a ** b يعني a أُس b

12-3 أساسيات البرمجة [2]

- **جملة for**: **جملة for:** بتكرر تنفيذ مجموعة أوامر: `for i in range(4):` وتحتها السطور **المزاجة بمسافات (indent)**
- **()range**
  - `range(end)` : من **0** لحد **end – 1**
  - `range(start, end)` : من **start** لحد **end – 1**
  - `range(start, end, increment)` : بتزيد بمقدار increment ومش بتوصل لـ end
  - علشان تعدّ من 1 لـ 10 لازم تكتب `range(1, 11)`
- **معاملات المقارنة**
  - `==` يساوي
  - `<` أقل من
  - `=<` أقل من أو يساوي
  - خد بالك: `=` تخصيص
- **if و else**: **التعبير الشرطي (Conditional expression):** بيرجّع صح (True) لو الشرط متحقق وخطأ (False) لو لأ.
- **if ~ elif ~ else**: بنستخدمها لما يبقى عندنا **أكتر من شرط**

for statement	جملة for: بتكرر السطور المزاحة تحتها
range()	بتحدد مدى قيم متغير التكرار (البداية، النهاية – 1، الزيادة)
Indentation	الإزاحة: مسافات قبل السطر بتحدد إنه جوه for أو if
Comparison operator	عامل المقارنة: بيقارن قيمتين (== != > < => =<)
Conditional expression	التعبير الشرطي: بيرجّع صح لو الشرط متحقق وخطأ لو لأ
if statement	جملة if: تفرّع حسب شرط
if ~ else	بتحدد اللي يحصل لو الشرط صح ولو غلط

if ~ elif ~ else	بتختبر أكتر من شرط بالترتيب

12-4 البرمجة التطبيقية [1]

- **القائمة (List): القائمة (List):** = مجموعة بيانات مترتبة ورا بعض
- **الوصول للعناصر:** `a[0]` أول عنصر (7)
- **إضافة عنصر بـ append:** `a.append(x)` بتضيف x في **آخر** القائمة
- **القائمة ثنائية الأبعاد: القائمة ثنائية الأبعاد (Two-dimensional list):** = قائمة جواها قوايم
- **برامج مشهورة على القوائم**
  - **المجموع:** نبدأ `total = 0` ونضيف `a[i]` لكل عنصر.
  - **أصغر قيمة:** نبدأ `m = a[0]` ولو `a[i] < m` نخلي `m = a[i]` .
  - **أكبر قيمة:** نفس الفكرة بـ `>` .
  - **العدّ:** `count = count + 1` كل ما عنصر يحقق الشرط.

List	القائمة: مجموعة بيانات مرتبة بالتسلسل تحت اسم واحد
Element	العنصر: كل قيمة موجودة في القائمة
Index	الفهرس: رقم مكان العنصر، ويبدأ من 0
append()	بتضيف عنصر في آخر القائمة
Two-dimensional list	قائمة ثنائية الأبعاد: صفوف وأعمدة، والعنصر a[i][j]
a[i]	العنصر اللي فهرسه i في القائمة a

12-5 البرمجة التطبيقية [2]

- **يعني إيه دالة؟: الدالة (Function):** = مجموعة أوامر متجمعة في وحدة واحدة ليها اسم
- **التعريف والاستدعاء:** الشكل: `def name(arg1, arg2):` وتحتها الأوامر مزاحة
- **دالة بأكتر من وسيطة:** `def area(w, h):` / `return w * h` ثم `area(4, 3)` → 12
- **أنواع الدوال**
  - **دوال مدمجة (Built-in functions):** جاهزة من غير تعريف
  - **دوال معرّفة من المستخدم (User-defined functions):** إنت اللي بتكتبها بـ `def` .

Function	الدالة: مجموعة أوامر في وحدة واحدة ليها اسم
Argument	الوسيطة: المتغير اللي بيستقبل قيمة جوه الدالة
return	بترجّع نتيجة الدالة للمكان اللي ناداها
Return value	قيمة الإرجاع: النتيجة اللي الدالة بترجّعها
Function call	استدعاء الدالة: تشغيلها باسمها وقيم الوسائط
def	الكلمة اللي بنعرّف بيها دالة في بايثون
Built-in function	دالة مدمجة: جاهزة من غير تعريف زي print()
User-defined function	دالة معرّفة من المستخدم: بنكتبها بـ def

ذاكر المادة كلها مجاناً

شرح كل الدروس، أسئلة بالإجابات، بنك على نمط الوزارة، وامتحان بشكل الوزارة

zerooneacademy.com

مهندس رحيم