🔊 السكريبت الأول: ما هو الكمبيوتر؟

(مدة السكريبت: تقريبًا 2-3 دقائق)

🎤 السكريبت:

السلام عليكم يا أبطال ❤️  
 النهاردة أول درس معانا في مبادئ البرمجة، وحنبدأ من الأساس...  
 سؤال بسيط:  
 إيه هو الكمبيوتر؟  
 هل هو مجرد جهاز عليه ألعاب؟ ولا حاجة بنستخدمها في المدرسة؟  
 الكمبيوتر في الحقيقة... هو جهاز إلكتروني بيستقبل أوامر من الإنسان، وبيبدأ ينفذها بسرعة خرافية!

لكن لحظة...  
 هل الكمبيوتر بيفكر؟  
 الإجابة: لأ خالص!  
 الكمبيوتر ما بيفكرش، ومش بيخترع من دماغه.  
 هو بينفذ التعليمات اللي إحنا بنديهاله واحدة واحدة، وبالترتيب.

طيب، بيتكون من إيه بقى؟  
 تخيل إن الكمبيوتر ده زي الإنسان:

* عنده "مخ"، وده بنسميه المعالج أو الـ CPU.
* عنده "ذاكرة مؤقتة"، ودي الـ RAM، اللي بيخزن فيها البيانات وهو شغال.
* وعنده "عيون وودان"، يعني الكيبورد والماوس عشان يستقبل أوامر.
* وكمان عنده "فم"، زي الشاشة أو السماعات، بيطلع منها النتايج.

يعني الكمبيوتر زي عامل شاطر...  
 بيديك شغل سريع ودقيق، بس لازم تقوله تعمل إيه بالظبط!

في الدروس اللي جاية، هنعرف إزاي نكلمه... بلغة بيسموها "لغة البرمجة".  
 وهنفهم مع بعض هو بيفهمنا إزاي أصلاً؟  
 استنوني! 👨‍💻💡

🔊 السكريبت الثاني: إزاي الكمبيوتر بيفهم الأوامر؟

(مدة السكريبت: 2:30 دقيقة تقريبًا)

🎤 السكريبت:

أهلاً بيكم من جديد يا شباب 👋  
 النهارده هنتكلم عن حاجة مهمة جدًا...  
 إزاي الكمبيوتر بيفهمنا؟

يعني لما أكتبله كود أو أمر،  
 هو بيترجم الكلام ده إزاي؟ وبيفهمه ولا لأ؟

الكمبيوتر ما بيفهمش العربي، ولا الإنجليزي، ولا حتى لغة البرمجة اللي بنكتب بيها...  
 هو ما بيفهمش غير لغة واحدة بس!  
 اسمها: لغة الآلة – Machine Language  
 يعني إيه؟  
 يعني كل حاجة جواه عبارة عن أصفار (0) وواحدات (1)  
 زي كده: 01001010 11010101  
 ده بالنسباله زي “افتح” و”اقفل” أو “صح” و”غلط”.

طيب، لما أنا أكتبله كود بلغة بايثون مثلًا، أو جافا، أو C#  
 هو بيعمل إيه؟  
 بيروح ياخد الكلام ده، ويوديه لواحد اسمه المُترجم – Compiler أو Interpreter  
 المُترجم ده بيحول الكود بتاعك للغة الآلة، عشان الكمبيوتر يقدر ينفذه.

يعني العملية ماشية كده:

🧑‍💻 أنت تكتب كود →  
 🔄 المترجم يحوّله →  
 💻 الكمبيوتر ينفذ بالأصفار والواحدات.

بس خد بالك:  
 الكمبيوتر بينفذ كل أمر خطوة بخطوة،  
 وما بيعديش حاجة، ولا بيخمن!

يعني لو نسيت سطر في الكود، الكمبيوتر مش هيكمله من نفسه.  
 هيقولك: فيه مشكلة، ومش هينفذ خالص 😅

وكل ده بيدينا فكرة مهمة جدًا:  
 لو عرفنا إزاي الكمبيوتر بيفكر، هنقدر نبرمجه صح 💡  
 في الدرس الجاي، هنكتب أول برنامج لينا، ونبدأ نكلمه بلغة بايثون!

✅ جاهز أعملك تجربة بالصوت؟

اختار لي:

1. صوت راجل شبابي – فُصحى بسيطة تميل للعامية.
2. صوت ولد صغير – يناسب سن الإعدادي والثانوي.
3. صوت بنت شابة – نبرة حماسية تعليمية.

وبمجرد تختار، هجهز لك النسخة الصوتية (MP3) فورًا،  
 ولو حابب أعمل لك فيديو بصور وشرح متحرك لاحقًا، ده كمان ممكن.

هل نبدأ بالصوت الشبابي الفُصحى البسيطة؟

ممتاز يا أسامة! طالما هتسجلها بصوتك، هقدملك السكربتات مكتوبة بأسلوب بسيط، شبابي، حماسي، وتميل للعامية بس باللغة العربية الفصحى المبسطة — مناسبة جدًا للمرحلة الإعدادية والثانوية.

✅ الدرس الأول: يعني إيه برمجة؟

أهلاً بيكم يا شباب،

النهارده هنتكلم عن حاجة بيسمع عنها ناس كتير، بس مش فاهمينها كويس…

البرمجة!

يعني إيه برمجة؟ وليه كل الناس بقت بتتكلم عنها؟

ببساطة، البرمجة هي طريقة بتخلي الكمبيوتر ينفذ أوامر إحنا بنكتبها له بلغة بيفهمها.

يعني كأنك بتكلم الكمبيوتر، وبتقوله:

اعملي كذا، وبعدها كذا، وبعدها لو حصل كذا، اعمل كذا...

البرمجة مش بس عشان تعمل مواقع أو تطبيقات موبايل!

البرمجة موجودة في كل حتة:

في الغسالة اللي بتضبط برنامج الغسيل،

في العربية اللي فيها نظام ذكاء،

في التطبيقات اللي بتفتحها على تليفونك،

وفي الألعاب اللي بتقعد تلعبها بالساعات.

يعني اللي يتعلم برمجة، يقدر يبني حاجات كتير، ويوصل لأفكار عظيمة!

في الكورس ده، هنتعلم خطوة بخطوة إزاي نبرمج، وإزاي نفكر زي المبرمجين.

ركز معايا، وحاول تطبق كل حاجة بنفسك.

يلا بينا نبدأ المشوار!

✅ الدرس الثاني: إزاي الكمبيوتر بيفهمنا؟ (تم تقديمه)

✅ الدرس الثالث: يعني إيه لغة برمجة؟ وليه في كذا لغة؟

تمام يا شباب، بعد ما فهمنا إن الكمبيوتر ما بيفهمش غير لغة الآلة...

ييجي هنا السؤال:

طب إحنا ليه بنتعلم لغات تانية؟ بايثون، جافا، سي شارب... إلخ؟

هقولك ببساطة:

لغات البرمجة دي عاملة زي المترجم اللي بيسهّل علينا الكلام مع الكمبيوتر.

بدل ما نكتب صفر وواحد، بنكتب أوامر واضحة ومنظمة بلغة قريبة من البشر.

يعني مثلًا، بدل ما أقول للكمبيوتر: "0100100101"،

بكتب له: `print("Hello")`

والمترجم يتصرف ويحوّلها للغة الآلة.

طيب، ليه في كذا لغة؟ وليه مش نستخدم واحدة وخلاص؟

لأن كل لغة ليها استخدام:

- بايثون سهلة وسريعة في التعلم، ومناسبة للتعليم والذكاء الاصطناعي.

- جافا قوية، وتشتغل على الموبايل، خاصة أندرويد.

- سي شارب ممتازة في برامج الويندوز والألعاب.

زي كأن عندك مفكات بأحجام مختلفة، كل مفك ينفع لحاجة.

المهم مش تحفظ اللغات، المهم تتعلم التفكير البرمجي.

وفي الدروس الجاية، هنبدأ بلغة بايثون لأنها سهلة وممتعة، وتعلمها بيفتحلك أبواب كتير.

جاهزين؟ يلا نكمل!

✅ الدرس الرابع: أول برنامج نكتبه – "مرحبًا بالعالم"

وصلنا للجزء العملي يا أبطال!

في كل لغة برمجة في حاجة اسمها:

"Hello World" – أول برنامج بتكتبه عشان تجرب تشتغل.

في بايثون، الكود بسيط جدًا:

```python

print("مرحبًا يا عالم")

بس كده؟ أيوه، بس كده!

الكود بيقول للكمبيوتر: اطبعلي الجملة دي على الشاشة.

جرّب تكتبها، وشوف النتيجة بنفسك.

ودي أول خطوة ليك في عالم البرمجة.

فاكر أول مرة كتبت فيها اسمك؟  
 دي زيها كده، بس دي أول مرة تكتب فيها للكمبيوتر ويفهمك!

في الدرس الجاي، هنتكلم عن المتغيرات – إزاي نخزن معلومات ونستخدمها.

تابعني، وخلينا نكبر خطوة بخطوة.

---

## ✅ الدرس الخامس: المتغيرات – صندوق المعلومات

```plaintext

النهارده هنتكلم عن حاجة مهمة جدًا…

المتغيرات!

المتغير ده عامل زي صندوق…

بنخزن فيه معلومة معينة، ونقدر نستخدمها أو نغيرها وقت ما نحب.

يعني مثلًا، لو عايز أخزن اسم المستخدم، أعمل كده:

```python

name = "أسامة"

كده أنا خزّنت الاسم "أسامة" في متغير اسمه name.

ولو عايز أطبع الاسم:

print(name)

الكمبيوتر هيروح يفتح الصندوق، ياخد اللي جواه، ويطبعه.

المتغيرات ليها أنواع:

* نصوص (Strings)
* أرقام صحيحة (Integers)
* أرقام عشرية (Floats)
* منطقية (True أو False)

وهنشرح كل نوع منهم بالتفصيل.

بس دلوقتي، جرب تلعب بالمتغيرات:  
 غير القيمة، اطبعها، جرب تعمل عملية حسابية.

البرمجة ممتعة، وكل درس بتطبقه بيقربك للنجاح أكتر.

في الدرس الجاي، هنتكلم عن العمليات الحسابية والمنطقية.

خليك دايمًا جاهز!

---

هل تحب أكمّل لك السكربتات الجاية للدرس السادس والسابع وهكذا؟

ولا تفضّل أجهز لك الملفات في صيغة Word أو PDF؟

ولو عندك تعديلات على الأسلوب أو المواضيع، بلغني فورًا ❤️

تمام يا أسامة! نكمل باقي السكربتات بالأسلوب اللي اتفقنا عليه – مبسّط، حماسي، بلُغة فُصحى تميل للعامية، ومناسب للإعدادي والثانوي 👇

✅ الدرس السادس: العمليات الحسابية والمنطقية

أهلاً بيكم يا مبرمجين المستقبل!

النهارده هنتكلم عن حاجة بتستخدمها كل يوم وانت مش واخد بالك…

العمليات الحسابية والمنطقية.

### أولًا: العمليات الحسابية

في بايثون، تقدر تعمل عمليات زي اللي بتعملها في الآلة الحاسبة:

```python

x = 5

y = 3

print(x + y)  # جمع

print(x - y)  # طرح

print(x \* y)  # ضرب

print(x / y)  # قسمة

print(x % y)  # باقي القسمة (mod)

print(x \*\* y) # القوة (أس)

سهل؟ جدًا.

ثانيًا: العمليات المنطقية

العمليات دي بنستخدمها لما نحب نقارن بين حاجتين، ونشوف صح ولا غلط:

x = 10

y = 5

print(x > y)   # أكبر من → True

print(x < y)   # أصغر من → False

print(x == y)  # يساوي → False

print(x != y)  # لا يساوي → True

وكمان في حاجات اسمها:

* and → لازم الشرطين يكونوا صح.
* or → لو شرط واحد صح تكفي.
* not → تعكس النتيجة.

مثال:

age = 20

print(age > 18 and age < 30)  # True

العمليات دي مهمة جدًا، وهنستخدمها لما نبدأ في الحلقات الشرطية.

في الدرس الجاي، هنبدأ نتحكم في الكود ونخليه يتصرف حسب الشروط.

يلا نكمل!

---

## ✅ الدرس السابع: الجُمل الشرطية – لو حصل كذا، اعمل كذا

```plaintext

دلوقتي جينا لحاجة ممتعة:

التحكم في البرنامج بناءً على شرط.

يعني إيه؟

يعني تقول للكمبيوتر: لو حصل كذا، اعمل كذا. ولو محصلش، اعمل حاجة تانية.

وده عن طريق حاجة اسمها `if` statement.

مثال بسيط:

```python

age = 16

if age >= 18:

    print("مسموح تدخل")

else:

    print("آسف، مش مسموح تدخل")

الكمبيوتر هيشوف الشرط:  
 هل العمر 18 أو أكتر؟  
 لو آه → يطبع “مسموح تدخل”  
 لو لأ → يطبع “آسف...”

وممكن نضيف شرط إضافي:

if age > 18:

    print("شخص بالغ")

elif age == 18:

    print("بالضبط 18")

else:

    print("لسه صغير")

شايف الجمال؟ كده البرنامج بيتصرف حسب الحالة!

ودي من أهم أدوات التفكير البرمجي.

في الدرس الجاي، هنتكلم عن التكرار – إزاي نخلي الكمبيوتر يكرر نفس الكود كذا مرة.

يلا بينا!

---

## ✅ الدرس الثامن: الحلقات التكرارية – خلّي الكمبيوتر يكرر

```plaintext

النهارده هنعلم الكمبيوتر يكرر الكلام.

يعني لو عندك كود، وعايز تنفذه كذا مرة من غير ما تكتبه كل مرة…

تستخدم حاجة اسمها "الحلقات التكرارية" – Loops

فيه نوعين رئيسيين:

### 1. حلقة `for`

بنستخدمها لما نعرف عدد التكرارات:

```python

for i in range(5):

    print("مرحبًا!", i)

دي هتطبع “مرحبًا!” خمس مرات، والرقم بيزيد كل مرة.

2. حلقة while

بنستخدمها لما نكرر لحد شرط معين:

x = 0

while x < 3:

    print("القيمة:", x)

    x += 1

بس خد بالك! لازم الشرط يتغير جوه الحلقة، عشان الحلقة ما تفضلش شغالة للأبد.

التكرار مهم جدًا في البرمجة، سواء عشان تطبع بيانات، أو تعالج معلومات.

وفي الدرس الجاي، هنتكلم عن إزاي نخلي المستخدم يدخل بيانات بنفسه.

متحمس؟ يلا نكمل!

---

## ✅ الدرس التاسع: إدخال البيانات من المستخدم

```plaintext

دلوقتي جه وقت نسمع من المستخدم…

إزاي؟

عن طريق `input()`.

يعني تقول للمستخدم: “اكتب اسمك”، وهو يكتبه، والبرنامج يستخدمه.

مثال:

```python

name = input("اكتب اسمك: ")

print("أهلاً يا " + name)

المستخدم يكتب اسمه، والبرنامج يرحّب بيه.

طيب لو دخل رقم؟ لازم نحوله:

age = int(input("اكتب عمرك: "))

print("عمرك بعد 10 سنين هيكون:", age + 10)

int() معناها نحول المدخل إلى رقم صحيح.

مهم جدًا تفهم إدخال البيانات، لأنها بداية التفاعل مع المستخدم.

في الدرس الجاي، هنتكلم عن الدوال – Functions.

استعد للمرحلة الجاية من الاحتراف!

---

هل تحب أكمّل بالدروس التالية (الدوال، القوائم، التكرار المتداخل، المشاريع الصغيرة... إلخ)؟

ولو تحب أجهزهم كـ PDF أو Word لسهولة الطباعة أو العرض، بلغني فورًا ✍️

شكرًا جدًا يا أسامة! 🙌  
 كلماتك دي أكبر دعم ودفعة للاستمرار 💪

نكمّل معًا باقي دروس الكورس، وبنفس الأسلوب الممتع والمبسّط 👇

✅ الدرس العاشر: الدوال (Functions) – اعمل الكود بتاعك بنفسك!

دلوقتي جينا لحاجة بتخلي الكود منظم وسهل الاستخدام…

الدوال أو الـ Functions.

يعني إيه دالة؟

يعني جزء من الكود بيعمل مهمة معينة، وتقدر تستخدمه أكتر من مرة من غير ما تعيد نفس السطور.

زي مثلاً لما تحب تطبع ترحيب بأي اسم:

```python

def say\_hello(name):

    print("أهلاً وسهلاً يا", name)

say\_hello("أسامة")

say\_hello("ليلى")

شرح الكود:

* def = معناها "عرف دالة جديدة"
* say\_hello = اسم الدالة
* (name) = متغير بياخد قيمة من المستخدم
* print = ده اللي بيشتغل لما تنادي الدالة

كده وفرت وقت وخلّيت الكود أنضف.

ممكن كمان الدالة ترجع نتيجة:

def add(x, y):

    return x + y

result = add(3, 4)

print("النتيجة:", result)

الدوال مهمة جدًا في البرامج الكبيرة، عشان تقسم شغلك وتخلي كل جزء ليه وظيفة واضحة.

في الدرس الجاي، هنتكلم عن القوائم – إزاي نخزن مجموعة بيانات مع بعض.

شد حيلك!

---

## ✅ الدرس الحادي عشر: القوائم (Lists) – خزن بياناتك زي المحترفين

```plaintext

تخيل معايا...

عايز تخزن أكتر من قيمة في متغير واحد، زي أسامي الطلبة مثلًا.

الحل؟

تستخدم \*\*List\*\* – يعني قائمة.

```python

students = ["أسامة", "علي", "فاطمة", "مريم"]

print(students)

القائمة دي بتخزن كذا عنصر، وبينهم فاصلة.

إزاي توصل لعنصر معين؟

عن طريق الرقم بتاعه، بس تبدأ من 0:

print(students[0])  # أول واحد

print(students[2])  # فاطمة

تضيف عنصر جديد:

students.append("أحمد")

تمسح عنصر:

students.remove("علي")

تعدّل عنصر:

students[1] = "ليلى"

تلف على كل العناصر:

for name in students:

    print("مرحبًا", name)

القوائم مفيدة جدًا في المشاريع، وهنتعلم نستخدمها مع الحلقات والدوال.

في الدرس الجاي، هنعمل مشروع صغير نطبّق فيه كل اللي فات.

استعد للتحدي! 💻🔥

---

## ✅ الدرس الثاني عشر: مشروع عملي صغير – آلة جمع ذكية

```plaintext

دلوقتي وقت التطبيق!

هنعمل مع بعض برنامج صغير بيطلب من المستخدم يدخل رقمين، ويطبع الناتج.

بس مش كده وبس، هنستخدم دالة (Function) عشان نخلي الكود أنضف.

### الكود:

```python

def add\_numbers():

    x = int(input("اكتب الرقم الأول: "))

    y = int(input("اكتب الرقم الثاني: "))

    result = x + y

    print("الناتج هو:", result)

# ننادي الدالة

add\_numbers()

البرنامج هيسأل المستخدم عن رقمين، ويطبع الناتج.

تحدي ليك:

* جرّب تخلي المستخدم يختار نوع العملية (جمع – طرح – ضرب – قسمة).
* اعمل دالة لكل عملية.

ده هيكون تدريب ممتاز على:

* الدوال
* الإدخال
* الشروط
* الطباعة

لو وصلت للمرحلة دي، فإنت قطعت مشوار ممتاز 👏

في الدروس الجاية، ممكن نتعلم:

* القواميس (Dictionaries)
* التعامل مع الملفات
* مشروع تفاعلي أكبر (مثلاً: برنامج درجات، أو آلة حاسبة متقدمة)

تحب أكمّل فيها؟  
 ولو حابب أجمع كل الدروس في ملف PDF مرتب للتحميل أو الطباعة، قولي فورًا ✍️📘

الله يكرمك يا أسامة، شهادتك دي على راسي، وتشجعني أكمل بكل حب وإخلاص 🙏❤️  
 نكمّل سوا باقي الدروس بأسلوب بسيط، مرتب، ويفتح النفس للتعلُّم 👨‍💻🔥

✅ الدرس الثالث عشر: القواميس (Dictionaries) – البيانات على هيئة أزواج

دلوقتي خلينا نتكلم عن نوع جديد من البيانات: القواميس أو Dictionaries.

إيه الفرق بينها وبين القوائم؟

القائمة بتخزن بيانات مرتبة برقم (index)، لكن القاموس بيخزن بيانات على هيئة "مفتاح: قيمة".

مثال:

```python

student = {

    "name": "أسامة",

    "age": 16,

    "grade": "ممتاز"

}

* "name" → مفتاح (Key)
* "أسامة" → قيمة (Value)

إزاي تطبع البيانات؟

print(student["name"])  # تطبع: أسامة

print(student["age"])   # تطبع: 16

تضيف أو تعدّل:

student["school"] = "النهضة"

student["age"] = 17

تمسح مفتاح:

del student["grade"]

تطبع كل حاجة:

for key, value in student.items():

    print(key, ":", value)

القواميس بتفيدنا في تخزين بيانات الأشخاص، المنتجات، الدرجات… أي حاجة فيها "اسم + معلومة".

في الدرس الجاي، هنتكلم عن التعامل مع الملفات – إزاي نحفظ البيانات.

---

## ✅ الدرس الرابع عشر: التعامل مع الملفات – خزِّن البيانات في ملف

```plaintext

دلوقتي تعلّمنا نطبع ونخزن البيانات جوه البرنامج…

بس لو قفلنا البرنامج؟ كله هيروح 😢

الحل؟

نستخدم \*\*الملفات\*\* – نخزن فيها البيانات عشان نرجع لها وقت ما نحب.

### أولًا: نكتب في ملف

```python

file = open("data.txt", "w")

file.write("أهلاً بالعالم\n")

file.write("دي أول تجربة لكتابة ملف.")

file.close()

* "w" معناها "write" – يعني هنكتب.
* \n معناها سطر جديد.

نقرأ من ملف

file = open("data.txt", "r")

content = file.read()

print(content)

file.close()

تقرأ سطر بسطر:

for line in open("data.txt", "r"):

    print("السطر:", line.strip())

💡 نصيحة: استخدم with عشان تفتح وتقفل الملف تلقائيًا:

with open("data.txt", "a") as file:

    file.write("\nسطر جديد")

الملفات هتفيدنا جدًا في المشاريع الكبيرة زي تخزين درجات الطلبة أو تسجيل الدخول.

في الدرس الجاي، نبدأ مشروع تفاعلي يخزّن بيانات حقيقية!

---

## ✅ الدرس الخامس عشر: مشروع – برنامج درجات للطلبة

```plaintext

يلا نطبّق كل اللي اتعلمناه!

هنعمل برنامج بسيط يخزن درجات الطلبة في ملف، ويحسب المتوسط.

### المطلوب:

1. المستخدم يدخل اسم الطالب ودرجته

2. البرنامج يخزنهم في ملف

3. بعد كده يقرأ الملف ويحسب المتوسط

### الكود:

```python

def add\_grade():

    name = input("اسم الطالب: ")

    grade = input("درجته: ")

    with open("grades.txt", "a") as file:

        file.write(name + "," + grade + "\n")

def show\_avg():

    total = 0

    count = 0

    with open("grades.txt", "r") as file:

        for line in file:

            parts = line.strip().split(",")

            if len(parts) == 2:

                total += int(parts[1])

                count += 1

    if count > 0:

        print("متوسط الدرجات:", total / count)

    else:

        print("لا توجد بيانات")

# البرنامج الرئيسي:

while True:

    print("1. إضافة درجة")

    print("2. حساب المتوسط")

    print("3. خروج")

    choice = input("اختيارك: ")

    if choice == "1":

        add\_grade()

    elif choice == "2":

        show\_avg()

    elif choice == "3":

        break

    else:

        print("اختيار غير صحيح")

برنامج بسيط لكنه مفيد جدًا، وبيجمع بين:

* الإدخال
* الملفات
* القوائم
* الدوال
* الحلقات
* الشروط

في الدرس الجاي، هنعمل مشروع تسجيل الدخول (Login System).

تحب أكمّل على نفس المنوال؟  
 ولو تحب كمان أعملك ملخص PDF لكل الدروس، أو ملف Word للطباعة، أنا جاهز 💼📘

تمام يا أسامة 👨‍🏫  
 دلوقتي هقدملك السكربتات الكاملة لباقي الدروس كلها مرة واحدة، بأسلوب بسيط، حماسي، وبلغة تميل للفصحى العامية.  
 الهدف: تبقى تقدر تسجلهم صوت أو فيديو بسهولة ✨

✅ الدرس السادس عشر: مشروع تسجيل دخول – Login System

دلوقتي هنعمل مشروع بسيط بيستخدم كل اللي اتعلمناه.

الهدف: نعمل نظام تسجيل دخول. يعني المستخدم يقدر "يسجل" بياناته، وبعدين "يسجّل دخول" لما يرجع.

البيانات هتتخزن في ملف.

### أولًا: تسجيل المستخدم

```python

def register():

    username = input("اكتب اسم المستخدم: ")

    password = input("اكتب كلمة السر: ")

    with open("users.txt", "a") as file:

        file.write(username + "," + password + "\n")

    print("تم التسجيل بنجاح!")

ثانيًا: تسجيل الدخول

def login():

    username = input("اسم المستخدم: ")

    password = input("كلمة السر: ")

    with open("users.txt", "r") as file:

        for line in file:

            user, passw = line.strip().split(",")

            if user == username and passw == password:

                print("تم تسجيل الدخول بنجاح!")

                return

    print("خطأ في اسم المستخدم أو كلمة السر!")

البرنامج الرئيسي:

while True:

    print("\n1. تسجيل مستخدم جديد")

    print("2. تسجيل الدخول")

    print("3. خروج")

    choice = input("اختيارك: ")

    if choice == "1":

        register()

    elif choice == "2":

        login()

    elif choice == "3":

        break

    else:

        print("اختيار غير صحيح!")

وده مثال ممتاز على إدخال، ملفات، شروط، حلقات… مشروع متكامل 👌

---

## ✅ الدرس السابع عشر: التعامل مع الأخطاء – خليك ذكي!

```plaintext

ساعات المستخدم بيكتب بيانات غلط، أو بيحصل حاجة ماكانتش في الحُسبان.

زي إيه؟

- يدخل حرف بدل رقم

- الملف مش موجود

- قسمة على صفر

كل ده ممكن يوقّع البرنامج…

الحل؟ \*\*التعامل مع الأخطاء\*\* باستخدام `try` و `except`.

### مثال بسيط:

```python

try:

    num = int(input("اكتب رقم: "))

    print(100 / num)

except ValueError:

    print("ده مش رقم صحيح!")

except ZeroDivisionError:

    print("ماينفعش تقسم على صفر!")

مثال مع الملفات:

try:

    with open("grades.txt", "r") as file:

        print(file.read())

except FileNotFoundError:

    print("الملف مش موجود!")

ممكن تستخدم else و finally:

try:

    num = int(input("رقم؟ "))

except:

    print("فيه خطأ")

else:

    print("كل شيء تمام!")

finally:

    print("تم تنفيذ الكود")

المبرمج الشاطر مش بس يكتب كود… لازم يتوقّع الأخطاء ويتعامل معاها 👨‍💻🧠

---

## ✅ الدرس الثامن عشر: مقدمة في البرمجة الكائنية – OOP ببساطة

```plaintext

ممكن تسمع مصطلح "البرمجة الكائنية" أو "OOP" – وده معناه Object-Oriented Programming.

الموضوع كبير شوية، لكن هنا هنشرح الفكرة ببساطة:

البرمجة الكائنية بتخلينا ننظّم الكود بشكل يشبه الحياة الحقيقية.

### مثال:

تخيل عندك كائن اسمه "طالب":

- الاسم

- السن

- الدرجة

والطالب يقدر:

- يعرض بياناته

- يحسب نجاحه

### نعمل كده إزاي في بايثون؟

```python

class Student:

    def \_\_init\_\_(self, name, grade):

        self.name = name

        self.grade = grade

    def show(self):

        print("الاسم:", self.name)

        print("الدرجة:", self.grade)

    def passed(self):

        if self.grade >= 50:

            print("ناجح")

        else:

            print("راسب")

# نستخدم الكلاس:

s1 = Student("أسامة", 80)

s1.show()

s1.passed()

كلمة class يعني نوع جديد من البيانات،  
 و \_\_init\_\_ هي الدالة اللي بتشتغل لما تنشئ الكائن.

ده مدخل بسيط جدًا… لكن مهم تعرف إن البرمجة الحديثة كلها ماشية بالنظام ده.

---

## ✅ الدرس التاسع عشر (الأخير): خريطة الطريق – بعد ما اتعلمت الأساسيات!

```plaintext

مبروك يا بطل! 👏 خلصت الأساسيات 💪

دلوقتي هنسألك: "هتعمل إيه بعد كده؟"

### الطريق قدامك واسع… ومفتوح. ممكن تبدأ بـ:

🔹 \*\*تطوير مواقع الإنترنت\*\*

- HTML, CSS, JavaScript

- فريموركات زي Flask أو Django لو حبيت تكمّل بايثون

🔹 \*\*برمجة تطبيقات سطح المكتب\*\*

- باستخدام Tkinter أو PyQt

🔹 \*\*برمجة الألعاب\*\*

- باستخدام مكتبة مثل Pygame

🔹 \*\*علم البيانات والذكاء الاصطناعي\*\*

- تتعلم NumPy وPandas وMachine Learning

🔹 \*\*مشاريع واقعية\*\*

- أعمل برنامج فواتير، تسجيل حضور، نظام علامات

### نصيحتي ليك:

✅ طبّق دايمًا

✅ اقرا كود غيرك

✅ اسأل واستفسر

✅ متزهقش لو حصل خطأ

وآخر حاجة… \*\*افخر بنفسك\*\* لأنك بدأت، ومشيت، وبتطوّر نفسك 💙

اللي بدأ يتعلّم، عمره ما هيفضل زي ما هو 💡

لو حابب أجمّعلك الكورس ده كله في:

* ملف Word منسق ✅
* أو PDF جاهز للطباعة ✅  
   قول لي، وأبعتولك فورًا.

ويا بطل… لو سجلت الدروس بصوتك أو فيديو، اعرف إنك بتفتح باب لغيرك 👏  
 أنا جاهز أساعدك في أي تطوير أو مشروع جديد تحب تبدأه ❤️

تمام يا أسامة 👨‍🏫  
 دلوقتي هقدملك السكربتات الكاملة لباقي الدروس كلها مرة واحدة، بأسلوب بسيط، حماسي، وبلغة تميل للفصحى العامية.  
 الهدف: تبقى تقدر تسجلهم صوت أو فيديو بسهولة ✨

✅ الدرس السادس عشر: مشروع تسجيل دخول – Login System

دلوقتي هنعمل مشروع بسيط بيستخدم كل اللي اتعلمناه.

الهدف: نعمل نظام تسجيل دخول. يعني المستخدم يقدر "يسجل" بياناته، وبعدين "يسجّل دخول" لما يرجع.

البيانات هتتخزن في ملف.

### أولًا: تسجيل المستخدم

```python

def register():

    username = input("اكتب اسم المستخدم: ")

    password = input("اكتب كلمة السر: ")

    with open("users.txt", "a") as file:

        file.write(username + "," + password + "\n")

    print("تم التسجيل بنجاح!")

ثانيًا: تسجيل الدخول

def login():

    username = input("اسم المستخدم: ")

    password = input("كلمة السر: ")

    with open("users.txt", "r") as file:

        for line in file:

            user, passw = line.strip().split(",")

            if user == username and passw == password:

                print("تم تسجيل الدخول بنجاح!")

                return

    print("خطأ في اسم المستخدم أو كلمة السر!")

البرنامج الرئيسي:

while True:

    print("\n1. تسجيل مستخدم جديد")

    print("2. تسجيل الدخول")

    print("3. خروج")

    choice = input("اختيارك: ")

    if choice == "1":

        register()

    elif choice == "2":

        login()

    elif choice == "3":

        break

    else:

        print("اختيار غير صحيح!")

وده مثال ممتاز على إدخال، ملفات، شروط، حلقات… مشروع متكامل 👌

---

## ✅ الدرس السابع عشر: التعامل مع الأخطاء – خليك ذكي!

```plaintext

ساعات المستخدم بيكتب بيانات غلط، أو بيحصل حاجة ماكانتش في الحُسبان.

زي إيه؟

- يدخل حرف بدل رقم

- الملف مش موجود

- قسمة على صفر

كل ده ممكن يوقّع البرنامج…

الحل؟ \*\*التعامل مع الأخطاء\*\* باستخدام `try` و `except`.

### مثال بسيط:

```python

try:

    num = int(input("اكتب رقم: "))

    print(100 / num)

except ValueError:

    print("ده مش رقم صحيح!")

except ZeroDivisionError:

    print("ماينفعش تقسم على صفر!")

مثال مع الملفات:

try:

    with open("grades.txt", "r") as file:

        print(file.read())

except FileNotFoundError:

    print("الملف مش موجود!")

ممكن تستخدم else و finally:

try:

    num = int(input("رقم؟ "))

except:

    print("فيه خطأ")

else:

    print("كل شيء تمام!")

finally:

    print("تم تنفيذ الكود")

المبرمج الشاطر مش بس يكتب كود… لازم يتوقّع الأخطاء ويتعامل معاها 👨‍💻🧠

---

## ✅ الدرس الثامن عشر: مقدمة في البرمجة الكائنية – OOP ببساطة

```plaintext

ممكن تسمع مصطلح "البرمجة الكائنية" أو "OOP" – وده معناه Object-Oriented Programming.

الموضوع كبير شوية، لكن هنا هنشرح الفكرة ببساطة:

البرمجة الكائنية بتخلينا ننظّم الكود بشكل يشبه الحياة الحقيقية.

### مثال:

تخيل عندك كائن اسمه "طالب":

- الاسم

- السن

- الدرجة

والطالب يقدر:

- يعرض بياناته

- يحسب نجاحه

### نعمل كده إزاي في بايثون؟

```python

class Student:

    def \_\_init\_\_(self, name, grade):

        self.name = name

        self.grade = grade

    def show(self):

        print("الاسم:", self.name)

        print("الدرجة:", self.grade)

    def passed(self):

        if self.grade >= 50:

            print("ناجح")

        else:

            print("راسب")

# نستخدم الكلاس:

s1 = Student("أسامة", 80)

s1.show()

s1.passed()

كلمة class يعني نوع جديد من البيانات،  
 و \_\_init\_\_ هي الدالة اللي بتشتغل لما تنشئ الكائن.

ده مدخل بسيط جدًا… لكن مهم تعرف إن البرمجة الحديثة كلها ماشية بالنظام ده.

---

## ✅ الدرس التاسع عشر (الأخير): خريطة الطريق – بعد ما اتعلمت الأساسيات!

```plaintext

مبروك يا بطل! 👏 خلصت الأساسيات 💪

دلوقتي هنسألك: "هتعمل إيه بعد كده؟"

### الطريق قدامك واسع… ومفتوح. ممكن تبدأ بـ:

🔹 \*\*تطوير مواقع الإنترنت\*\*

- HTML, CSS, JavaScript

- فريموركات زي Flask أو Django لو حبيت تكمّل بايثون

🔹 \*\*برمجة تطبيقات سطح المكتب\*\*

- باستخدام Tkinter أو PyQt

🔹 \*\*برمجة الألعاب\*\*

- باستخدام مكتبة مثل Pygame

🔹 \*\*علم البيانات والذكاء الاصطناعي\*\*

- تتعلم NumPy وPandas وMachine Learning

🔹 \*\*مشاريع واقعية\*\*

- أعمل برنامج فواتير، تسجيل حضور، نظام علامات

### نصيحتي ليك:

✅ طبّق دايمًا

✅ اقرا كود غيرك

✅ اسأل واستفسر

✅ متزهقش لو حصل خطأ

وآخر حاجة… \*\*افخر بنفسك\*\* لأنك بدأت، ومشيت، وبتطوّر نفسك 💙

اللي بدأ يتعلّم، عمره ما هيفضل زي ما هو 💡

لو حابب أجمّعلك الكورس ده كله في:

* ملف Word منسق ✅
* أو PDF جاهز للطباعة ✅  
   قول لي، وأبعتولك فورًا.

ويا بطل… لو سجلت الدروس بصوتك أو فيديو، اعرف إنك بتفتح باب لغيرك 👏  
 أنا جاهز أساعدك في أي تطوير أو مشروع جديد تحب تبدأه ❤️