



البرمجة والذكاء الاصطناعي

PROGRAMMING & ARTIFICIAL INTELLIGENCE

تعلم البرمجة اليوم ... واصنع المستقبل غداً 🚀

أمثلة برمجية وتطبيقية تفاعلية بأسلوب شيق وجذاب يناسب الطلاب



شرح مبسط ومبتكر لكل درس مدعوم بالأمثلة البصرية والرسوم التوضيحية



بناء مهارات التفكير المنطقي والخوارزمي وحل المشكلات الحياتية



تدريبات وتحديات برمجية ذكية تصقل مهاراتك خطوة بخطوة نحو التميز



إعداد وإشراف نخبة التعليم والتطوير:

المهندس / السيد سامي أفندي

منصة سيد أفندي التعليمية | Sayed Afandy Academy

📞 01030745596

امسح الكود ضوئياً
لزيارة المنصة التعليمية
sayedafandy.com



ARTIFICIAL INTELLIGENCE DATABASE ALGORITHMS WEB DEV JAVASCRIPT PYTHON
PROJECTS

الدرس الأول

كود الكتابة في المحرر (Console Output)



أبسط مثال لطباعة نص داخل المحرر ووحدة التحكم:

example1.js

console.log("Hello");

الأمر `console.log()` هو المسؤول عن إخراج النصوص وعرض الرسائل داخل وحدة التحكم (Console).



مثال 2: التعليقات البرمجية

(Comments)

لو في كلمة أو جملة عايزها ماتنفذش وتكون مجرد توضيح نضع `//` في أول الجملة:

comment.js

// Greetings for the day.

لا يظهر أي ناتج لأن المحرر يتجاهل التعليقات تماماً



مثال 1: كيف نطبع Good Morning؟

greet.js

console.log("GoodMorning");

GoodMorning

الناتج ⚡

ثانياً: المسائل والعمليات الرياضية

كتابة مسألة رياضية وظهور الناتج مباشرة في وحدة التحكم نكتب العملية بدون علامات اقتباس:

4

الناتج يكون: ⚡

math.js

console.log(2 + 2);

احذر الوقوع هنا!

الصواب: `console.log(2 + 2);` → يطبع 4

فخ الامتحان: فخ علامات الاقتباس

الخطأ: `console.log("2 + 2");` → يطبع "2 + 2"

السبب البرمجي: وجود علامات الاقتباس " " يجعل المتصفح يطبع النص حرفياً، وبدونها يحسب العملية الرياضية فوراً!

العمليات الحسابية الأربع الأساسية (The Four Basic Operations)

الطرح ، القسمة ، الضرب ، الجمع تسمى بـ العمليات الحسابية الأربع.

القسمة (/)

الضرب (*)

الطرح (-)

الجمع (+)

الدرس الثاني

طرق الحساب (1) - العمليات الحسابية الأربع



العمليات الحسابية الأربع (The Four Operations):

العمليات الحسابية الأربع الأساسية هي: **الجمع** ، **الطرح** ، **الضرب** ، **والقسمة**.
المعاملات الحسابية (Operators): هي الرموز المستخدمة في البرمجة لتنفيذ الحسابات الرياضية:

/ قسمة

* ضرب

- طرح

+ جمع

علامة +

+ أولاً: عملية الجمع (Addition)

نستخدم علامة الزائد (+) لإجراء عملية الجمع. اجعل الكمبيوتر يحسب $345 + 432$:

777

⚡ النتائج في وحدة التحكم:

addition.js

console.log(432 + 345);

علامة -

- ثانياً: عملية الطرح (Subtraction)

نستخدم علامة الناقص (-) لإجراء عملية الطرح. اجعل الكمبيوتر يحسب $100 - 50$:

50

⚡ النتائج في وحدة التحكم:

subtraction.js

console.log(100 - 50);

احذر الوقوع هنا!

⚡ فخ الامتحان: علامات الآلة الحاسبة الممنوعة

✓ الصواب: console.log(100 * 5 / 2); → يطبع 250

✗ الخطأ: console.log(100 × 5 ÷ 2); → خطأ SyntaxError

💡 السر البرمجي: في البرمجة نستخدم النجمة * للضرب والشرطة / للقسمة؛ ورموز الرياضيات الورقية (× و ÷) يرفضها محرك الكود!

⚠ تنبيه هام وملاحظة ذهبية للمبرمج الصغير:

الضرب والقسمة في البرمجة يستخدمان رموزاً مختلفة عن الرياضيات المعتادة في الكراس:

- نستخدم رمز النجمة (*) لعملية الضرب (وليس علامة ×).
- نستخدم الشرطة المائلة (/) لعملية القسمة (وليس علامة ÷).

الدرس الثالث

التجميع (1) - ربط النصوص مع العمليات الحسابية



تأمل هذه الأسطر الثلاثة المنفصلة (الطريقة التقليدية):

النتائج منفصلة في 3 أسطر:

```
Set of 2 coffees
500
codycoins
```

separate.js

```
console.log("Set of 2 coffees");
console.log(250 * 2);
console.log("codycoins");
```



دمج النصوص بالعمليات الحسابية في سطر واحد (Concatenation)

نستخدم علامة الزائد (+) لربط النصوص مع الصيغ الرياضية في سطر واحد وأنيق ومختصر:

القاعدة العامة للدمج:

syntax.js

```
console.log("character " + (formula) + " character");
```

★ نفس النتيجة السابقة ولكن في سطر برمجي واحد فقط:

combined.js

```
console.log("Set of 2 coffees " + (250 * 2) + " codycoins");
```

Set of 2 coffees 500 codycoins

⚡ النتيجة في سطر واحد:

احذر الوقوع هنا!

⚡ فخ الامتحان: فخ دمج الأرقام مع النصوص

الصواب: ☒ console.log("المجموع: " + (20 + 10) + "); → "المجموع: 30"

✗ الخطأ: console.log("المجموع: " + (20 + 10) + "); → "المجموع: 1020"

💡 السر البرمجي: علامة + مع النصوص مغناطيس يدمج الكلمات، ولإجراء الجمع الحسابي أولاً ضع العملية دائماً داخل أقواس () منفصلة!

📌 فائدة برمجية ذكية:

بدل أن نكتب 3 أوامر طباعة منفصلة ونستولك أسطراً ومساحة، استطعنا دمج الجملة مع الناتج في أمر `console.log()` واحد فقط! هذا يجعل كودك أسرع وأنظف واحترافياً.

الدرس الرابع

التجميع (2) - باقي القسمة وأولويات العمليات ⚡



باقي القسمة (Modulus Operator %):

علامة (%) في لغات البرمجة لا تعني النسبة المئوية، بل تعني باقي القسمة الصحيح:

1

باقي القسمة: ⚡

modulus.js

console.log(10 % 3);

التوضيح: $10 \div 3 = 3$ والباقي 1 (لأن $3 \times 3 = 9$ ، و $10 - 9 = 1$).



هرم أولويات العمليات الحسابية (Order of Precedence)

الأولوية 3

الجمع + والطرح -

الأولوية 2

الضرب * والقسمة /

الأولوية 1

الأقواس ()

الحالة 2: مع الأقواس

p2.js

console.log((2 + 3) * 4);

20

الناتج: ⚡

الأقواس أولاً: $5 = (3+2)$ ، ثم $20 = 4*5$

الحالة 1: بدون أقواس

p1.js

console.log(2 + 3 * 4);

14

الناتج: ⚡

ضرب $3*4 = 12$ ، ثم $14 = 2 + 12$

سر باقي القسمة % للأصغر

إذا كان الرقم الأول أصغر من الثاني، فالناتج هو الرقم الأول نفسه دائماً: $3 \% 10 = 3$ و $5 \% 8 = 5$!

تساوي الأولويات (من اليسار لليمين)

عند اجتماع الضرب مع القسمة أو الجمع مع الطرح، ينفذ الحاسوب العملية التي تأتي أولاً من جهة اليسار: $15 = 3 * 2 / 10$



تحدي المبرمج الذكي:

ما هو الناتج المتوقع لتنفيذ الأمر التالي في لغة جافاسكريبت مع ذكر السبب؟

console.log(10 + 5 * 2 - (4 / 2));

الحل النموذجي: الأقواس أولاً ($4/2 = 2$) ثم الضرب ($5*2 = 10$) فتصبح: $18 = 2 - 10 + 10$.