

الأكاديمية العربية الدولية  
Arab International Academy

إسم المادة: مبادئ البرمجة

المحاضر: م. خليل محمد

---

الأكاديمية العربية الدولية - منصة أعد

# تعريف هامة

**البيانات** : هي الحقائق المجردة التي يتم تجميعها وتخزينها بواسطه نظام المعلومات .

المادة الخام الاولية للمعلومات ولها اشكال متعددة منها الارقام والحرروف والصور والاصوات ولقطات الفيديو والاسماء والتاريخ والاسعار ودرجات الحرارة ودرجات الاختبارات.

مثال : القطن عند جمعه من الحقل هو ماده خام اولية

المعلومات : هي بيانات تم معالجتها المعالجة هي اجراء بعض العمليات سواء الحسابية او المنطقية وتأخذ المعلومات اشكال عديدة مثل التقارير والجداول والقوائم والرسومات البيانية.

مثال : الغزل هو معلومة لأنه ناتج معالجه القطن الخام بيان فقد اجرينا عليه عملية وهي عملية الغزل ليتحول الخيوط

# تعريف هامة



نظام المعلومات: هو نظام يجمع بين التكنولوجيا مثله في الكمبيوتر والافراد ويسمح لاي مؤسسه بجمع وتخزين البيانات ثم استخلاص معلومات من هذه البيانات بعد ذلك

البرنامـج: مجموعة من الاوامر مكتوبة بشكل معين تخبر الكمبيوتر بما هو مطلوب منه

البرمـجة: هي عملية انشاء البرامـج . الاوامر المكتوبة بها البرنامـج او هي طريـقة حل المسائل تهدف لتقديم الحل على صورة خطوات مرتبة ترتيبا منطقـية اذا تبعـنا نصل الى الحل المسـالة العمـليـات التي يقوم بها الكمبيوتر

# خطوات تحويل البيانات إلى معلومات

## 1- ادخال البيانات (قراءة البيانات):

يقوم الكمبيوتر بمعالجه البيانات لتحويلها لمعلومات فلا بد من ادخال هذه البيانات الى الكمبيوتر باحدى وحدات الادخال ويتم ادخال البيانات بواسطة لوحة المفاتيح او الفارة او القرص المرن او المدمج او اي وسيلة ادخال أخرى.

## 2- تخزين البيانات في الذاكرة الداخلية:

- عند ادخال البيانات للكمبيوتر فإنه يتم تخزينها بالذاكرة الداخلية ويخصص لكل مكان في الذاكرة يحمل بيانات اسماء محددة
- عن طريق هذه الاسماء يستطيع البرنامج التعامل مع البيانات الموجودة مع بالذاكرة
- ولأنه قد تتغير القيم المخزنة في اي مكان بالذاكرة نتيجة تخزين قيم مختلفة فيها خلال البرنامج الواحد فان الاماكن المختلفة بالذاكرة تسمى ..المتغيرات..

## 3- اجراء العمليات الحسابية على البيانات :

يقوم الكمبيوتر بواسطة المعالج الدقيق بتنفيذ بعض العمليات الحسابية على البيانات المتاحة مثل الجمع والطرح والضرب وغيرها .....

# خطوات تحويل البيانات الى معلومات



4 تتم هذه العمليات حسب الاوامر التي يتم تغذية الكمبيوتر بها  
5 تكرار تنفيذ امر معين او مجموعه من الاوامر عده مرات:

- الكمبيوتر جهاز لا يكل ولا يمل لذلك يمكنه تكرار جزء او اكثرب من البرنامج اي عدد من المرات على ان يوجد شرط التكرار فاذا ما تحقق هذا الشرط استمر التكرار واذا اصبح غير متحقق انتهت عملية التكرار
- عملية ادخال البيانات اسماء الطلاب مدرسة ما مثل جيد لكرار.

## 6 اخراج النتائج:

عند انتهاء معالجه البيانات وتحويلها لمعلومات تصبح المعلومات جاهزة الارجاع على احدى وحدات الارجاع مثل الشاشة الطابعة او تخزينها على اقراص او غيرها

# تعريف:

الخوارزمية algorithm: سلسلة الخطوات التي تتبع بالكمبيوتر لحل مشكله معينة.

لغات البرمجة: الكمبيوتر لا يفهم اللغة الانجليزية او العربية او اي لغة اخرى انه يفهم فقط لغة الالة، لغة الالة : تكون من رقمين فقط هما 0 ، 1

لغة البرمجة: هي لغة تكتب بالحروف الانجليزية ولكن بقواعد مختلفة تتغير من لغة لغة أخرى.

ولكل لغة برمجه برنامج خاص بها يسمى المترجم او المفسر بحسب اللغة ( يقوم بتحويلها الى لغة الالة التي يفهمها الكمبيوتر )

مثال: اذا اردت الحديث مع شخص ياباني فانك تحتاج بمتراجم من اليابانية الى العربية واذا اردت الحديث مع الشخص فرنسي فانك تستعين بمتراجم من الفرنسية وهكذا .

- توجد العديد والعديد من لغات البرمجة مثل لغة الفورتران ولغة C++ ولغة الجافا ولغة البيزك التي يوجد منها العديد مثل الاصدار G-W –Basic الذي يعمل تحت نظام الدوس DOS اما Visual Basic فهو يعمل تحت نظام التشغيل widows

# المترجم Compiler والمفسر Interpreter

ما هو **المترجم Compiler** المترجم هو برنامج كمبيوتر يحول الكود (التعليمات البرمجية) المكتوب بلغة برمجة عالية المستوى إلى كود الآلة، وهو برنامج يقوم بترجمة الكود الذي يمكن قراءته من قبل الإنسان إلى لغة يفهمها معالج الكمبيوتر (ثنائي 1 و 0 بت)، يعالج الكمبيوتر كود الآلة لأداء المهام المقابلة.

يجب أن يلتزم المترجم بقاعدة بناء الجملة الخاصة بلغة البرمجة التي كتب بها، مع ذلك، فإن المترجم ليس سوى برنامج ولا يمكنه إصلاح الأخطاء الموجودة في هذا البرنامج. لذلك، إذا قمت بخطأ ما، فأنت بحاجة إلى إجراء تغييرات في بناء جملة برنامجك. خلاف ذلك، لن يتم تجميعها.

## ما هو **المفسر Interpreter**

المفسر أو المترجم الفوري هو برنامج كمبيوتر والذي يحول كل جملة في البرنامج عالي المستوى إلى كود الآلة، يتضمن الكود المصدري تعليمات البرمجية المجمعة مسبقاً والنصوص. يقوم كل من المترجم والمفسر بنفس المهمة (وهي تحويل لغة البرمجة عالية المستوى إلى كود الآلة). مع ذلك، سيقوم المترجم بتحويل الكود إلى كود آلة (أي إنشاء ملف `exe` قبل تشغيل البرنامج، لكن يقوم المفسر بتحويل الكود إلى كود الآلة عند تشغيل البرنامج).

# الفرق بين المترجم Compiler والمفسر Interpreter

تم المقارنة بين المترجم والمفسر بناء على عدة أسس منها:

- |                    |                          |
|--------------------|--------------------------|
| 1. خطوات البرمجة   | .8. متطلبات الذاكرة      |
| 2. الميزات والعيوب | .9. الأنسب               |
| 3. كود الآلة       | .10. تحسين الكود         |
| 4. وقت التشغيل     | .11. الكتابة الديناميكية |
| 5. النموذج         | .12. الاستعمال           |
| 6. توليد البرنامج  | .13. تنفيذ الخطأ         |
| 7. التنفيذ         | .14. الأخطاء             |
|                    | .15. اللغات المستخدمة    |

# وظائف المترجم Compiler والمفسر Interpreter

## وظائف المفسر:

1. يقوم المفسر بتحويل الكود المصدر سطراً بسطراً أثناء وقت التشغيل.
2. المفسر يترجم برنامجاً مكتوباً بلغة عالية المستوى إلى لغة على مستوى الآلة.
3. يسمح المفسر بتقييم البرنامج وتعديله أثناء تنفيذه.
4. يتم قضاء وقت أقل نسبياً في تحليل البرنامج ومعالجته.

## وظائف المترجم:

1. يقرأ المترجم الكود المصدري، ويخرج الكود القابل للتنفيذ.
2. يترجم البرامج المكتوبة بلغة عالية المستوى إلى تعليمات يستطيع الكمبيوتر فهمها.
3. فهو يحول النص الذي يكتبه المبرمج إلى تنسيق يمكن أن تفهمه وحدة المعالجة المركزية.
4. عملية التجميع معقدة نسبياً، يقضي الكثير من الوقت في تحليل البرنامج ومعالجته.
5. النتيجة القابلة للتنفيذ هي شكل من أشكال الكود الثنائية الخاصة بالجهاز.

# ما هي البرمجة؟



البرمجة وتعرف بالإنجليزية (programming) وهي عبارة عن الوسيلة التي يقوم بها المبرمجين بكتابة تعليمات وتوجيهه مجموعة من الأوامر، أو ما يعرف بالشيفرة أو الكود (code) لجهاز الحاسوب؛ لتوجيهها وإعلامه بكيفية التعامل مع البيانات، وكيفية تنفيذ سلسلة من الأعمال المطلوبة تسمى خوارزمية

## ما هي أسباب ابتكار لغات البرمجة؟

يعود السبب في ذلك وبكل بساطة أن لغتنا تحتوي على عبارات وجمل يمكن فهمها بعده معاني، ولكن جهاز الحاسوب مثل أي جهاز كهربائي، لا يفهم إلا لغة الآلة التي تعتمد على نظام العد الثنائي (0,1) المناظرة لحالي الإيقاف والتشغيل (On-Off) والتي تعني إيقاف أو تشغيل الآلة من قبل الإنسان. لذا فقد وضعت له لغات تشبه لغة الرياضيات حيث أن لعباراتها معنى واحد فقط.

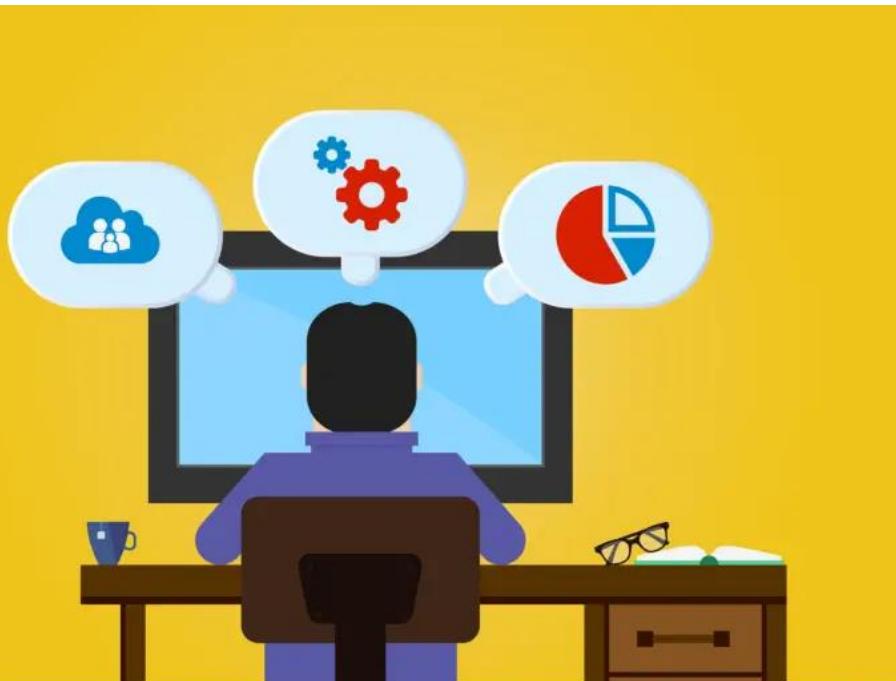
# لغة البرمجة:

**لغة البرمجة** بالإنجليزية **Programming language** : تُعد لغات البرمجة البرمجة الوسيط بين الأشخاص و جهاز الحاسوب، وهي عبارة عن لغة خاصة يستخدمها المبرمجين لكتابه البرمجيات، النصوص أو مجموعة من التعليمات والأوامر المُرتبة ترتيب منطقي، تكتب بناءً على مجموعة من القواعد؛ لإرشاد الحاسوب لتنفيذ مهام محددة، والتي تُحدد بحسب كل لغة من لغات البرمجة، وتكون مكتوبة على شكل كودات (Code) باستخدام لغات البرمجة أي لغات جهاز الحاسوب، ومن ثم تُمرر هذه الأوامر والتعليمات بمجموعة من المراحل إلى أن يتم تنفيذها على جهاز الحاسوب.

## كيف تمكنا لغات البرمجة من التواصل مع جهاز الحاسوب؟

لغات البرمجة تمكّنا من التواصل مع جهاز الحاسوب من خلال تقديم مجموعة من التوجيهات والتعليمات التي يمكن له أن يفهمها وينفذها، ولغة البرمجة كاللغات البشرية لها أسس وقواعد يج.ب علينا أن نتبعها وأن نتّقيد بها

# لغات البرمجة منخفضة المستوى وعالية المستوى



## لغات المستوى المنخفض :

مثل لغة الاله ولغة التجميع ولا يحتاج المفسر او المترجم ليفهما الكمبيوتر وهي لغات صعبة ومعقدة الاستخدام بالنسبة للإنسان

## لغات المستوى العالي :

اسهل في تعليمها وفهمها لأنها اقرب للغة الانجليزية العادية فهي تتكون من عبارات لغوية في شكل كلمات وتعابيرات وهي تحتاج لمترجم او مفسر ليحولها للغة الاله حتى يمكن للكمبيوتر فهما وتنفيذها.

# لغات البرمجة منخفضة المستوى



اللغات منخفضة المستوى : هي لغات قريبة على لغة الآلة ولا يفهمها إلا المتخصصون بعكس لغات البرمجة عالية المستوى، فهي تستخدم مفردات ورموز تُعد معقدة للمبرمج المبتدئ؛ لذلك يتم وصف اللغات ذات المستوى المنخفض في بعض الأحيان بأنها قريبة من لغة الأجهزة الإلكترونية .

## أقسام لغات البرمجة منخفضة المستوى:

تنقسم لغات البرمجة منخفضة المستوى في بعض الأحيان إلى فئتين، الجيل الأول والجيل الثاني.

### 1- الجيل الأول في لغة البرمجة المنخفضة المستوى:

يعتبر الجيل الأول من لغة البرمجة بشكل عام؛ لذلك يتم تسميته **GL1** أحد لغات الآلة (**machine Language**) تكتب هنا الأوامر على شكل سلسلة مكونة من الأرقام الثنائية (0، 1) حتى يفهمها الحاسب مباشرة وهي تقوم بعمليات بسيطة جدًا و تتعامل مباشرة مع ما هو مخزن في الذاكرة الرئيسية للجهاز.

# لغات البرمجة منخفضة المستوى

## 2- الجيل الثاني في لغة البرمجة المنخفضة المستوى :

يعرف الجيل الثاني منها باسم GL2، وهو ما يُسمى بلغة التجميع (Assembly Languages)

- ▶ تكتب الأوامر على شكل اختصارات باللغة الإنجليزية، تكون سهلة الفهم وتم ترجمة لغة التجميع للغة الآلة وتعامل لغة التجميع مع مكونات جهاز الحاسوب الداخلية بشكل مباشر.
- ▶ تجمع التعليمات البسيطة مباشرة إلى لغة الآلة بكل سهولة.
- ▶ بالرغم من أن لغة التجميع لا تخص المعالج الدقيق بشكل خاص إلا أنها تُعتبر إحدى لغات الجيل الثاني.

# لغات البرمجة عالية المستوى



لغات البرمجة عالية المستوى المعروفة باسم **high-level programming language**

تُعتبر نوع من أنواع لغات البرمجة، لا تتطلب من المبرمج الدخول إلى التفاصيل الخاصة التي

تعلق بتطبيق أوامر اللغة على مستوى الحاسوب، ويعُد هذا النوع من لغات البرمجة سهلاً في التعلم

والتطبيق بالنسبة للأشخاص المُبتدئين في البرمجة، حيث يسهل استخدامها بسبب مفرداتها وكلماتها

القريبة لغات التي يستخدمها الإنسان في الكلام (مثل اللغة الإنجليزية)، سهلة الاستخدام إلى حد ما

وسهل فهمها وتعديلها وراجعتها، كما أنها تعزل دلالات التنفيذ لهندسة الكمبيوتر عن مواصفات

البرنامج، مما يبسط التطوير.

# لغات البرمجة عالية المستوى

## أمثلة على اللغات عالية المستوى:

تنوع الأمثلة على اللغات عالية المستوى، ومنها:

■ **لغة بيسك (BASIC):** هي إحدى أبسط لغات البرمجة وأكثرها شعبية، تم تطويرها سنة 1964م.

■ **لغة السي (C)، وباللاتينية حرف C:** تُعد لغة السي من أفضل لغات البرمجة لكتابه برامجيات النظم، ومعظم نظم التشغيل مكتوبة بلغة السي مثل نظام windows.

■ **سي (C plus plus):** هي لغة مماثلة لغة السي (C) وتعتبر تكملة لها إلا أنها أكثر أماناً منها.

■ **سي شارب (#C):** هي لغة برمجة حديثة متعددة الأنماط، تم تطويرها في عام 2000م.

■ **لغة كوبول (COBOL):** هي اختصار (Common Business Oriented Language)، تُعد لغة كوبول لغة برمجة عالية المستوى.

■ **لغة جافا (Java):** لغة برمجة عالية المستوى تم ابتكارها عام 1992م.

# لغات البرمجة عالية المستوى

- **لغة Lisp:** هي اختصار لمصطلح (list processing) ، وتعني معالجة القوائم، وهي لغة برمجة وظيفية.
- **لغة البايثون :** (Python) لغة برمجة ذات مستوى عالي، تتميز هذه اللغة بسهولة كتابتها وقراءتها وسهولة تعلمها، وهي لغة قابلة للتطوير حيث استمر تطويرها إلى يومنا هذا.
- **دلفي :** (Delphi) تُعد لغة دلفي لغة من لغات البرمجة الناتجة عن لغة بascal، وهي لغة برمجة تُستخدم لتحديث البرامج والتطبيقات بشكل سريع وسهل.
- **لغة روبي :** (Ruby) لغة برمجية كائنية متعددة الاستخدامات، ظهرت لأول مرة في عام 1995م.
- **ادا (ADA)** هي لغة برمجة لها عدة استخدامات، صممها فريق، بطلب من وزارة الدفاع الأمريكية، وكانت اللغة التصميمية التي بدأت في الثمانينات .
- **لغة برولوج (Prolog)** هي عبارة عن لغة برمجة منطقية، صُمِّمت اللغة أساساً لكي تُستخدم في عمل برامج معالجة اللغات.

## الكود البرمجي: Code

هو نظام إشارات يستخدم لتمثيل الحروف أو الأرقام في إرسال رسائل التعليمات الموجودة في برنامج الكمبيوتر، غالباً ما تسمى التعليمات التي يكتبها المبرمج بلغة برمجة التعليمات المصدر، كما تسمى التعليمات التي تم تحويلها إلى لغة الآلة التي يفهمها الكمبيوتر رمز الجهاز أو التعليمات البرمجية القابلة للتنفيذ

يتكون الكود البرمجي من مجموعة من القواعد التي يتم إنشاءها بناءً على الهدف المراد تحقيقه من هذا الكود ومرجعه الأساسي (لغة البرمجة) حيث تعبر مجموعة من **expression** يتم تمثيله بأحرف ورموز معينة تسمى **syntax rule** والتي يتم تمثيلها بعدة شرائط مثل **CFG** أو **BNF** لتعقب داخل الأكواد و معرفة قواعد كتابة أي كود برمجي حيث يعتبر **syntax rule** شكل و تركيبة الجملة البرمجية و كذلك الجانب الآخر **semantic rule** و هي فهم ما المقصود و الهدف من كتابة هذا السطر بالحاسوب و له أنواع متنوعة مثل دراسة تأثير السطر البرمجي على الآلة، كل جملة برمجية لها تمثيل رياضي، كل جملة برمجية يحكمها شرط قبل و شرط بعد.

# الكود النظيف :Clen Code

ان بناء الكود النظيف هو أمر شخصي، وكل مطور لديه وجهة نظر شخصية، هناك بعض الأفكار التي تعتبر أفضل الممارسات والتي تشكل كودًا نظيفًا داخل العمل ومجتمع المبرمجين، لا يوجد اتفاق حاسم نهائياً، ولن يكون هناك على الإطلاق، لكن عندما نتحدث عن الكود النظيف، فإننا نتحدث عن أسلوب تطوير يركز على القارئ وينتج برامج يسهل كتابتها وقراءتها وصيانتها، والتعريف الأكثر شيوعاً للكود النظيف هو الكود الذي يسهل فهمه وتغييره، لكن ماذا يعني ذلك حقاً؟ دعنا نفصل هذه الجملة ونفحص النقاط الفردية وراءها:

- **سهولة الفهم :**تعني سهولة قراءة الكود، سواء كان ذلك القارئ هو الكاتب الأصلي للكود أو أي شخص آخر
- **التغيير السهل :**يعني أن الكود سهل التوسيع وإعادة البناء، كما أنه من السهل إصلاح الأخطاء في قاعدة الكود، ويمكن تحقيق ذلك إذا كان الشخص الذي يجري التغييرات يفهم الكود.

# أسباب أهمية الكود النظيف

## 1. الوضوح:

من السهل أن تنسى أن كل سطر يكتبه مطورو برامج الكود من المحتمل أن تتم قراءته عدة مرات بواسطة البشر خلال حياته، هؤلاء البشر عادة ما يكونون زملاء العمل، وهم مشغولون بإصلاح الأخطاء وإضافة الميزات لذلك يجب على كل مطور الاهتمام بالكود وجعله نظيف وواضح قدر الإمكان.

## 2. أفضل الممارسات:

في السنوات الأخيرة، نمت أفضل ممارسات البرامج مثل اختبار الوحدة، و اختبار يحركها التطوير (TDD) ، و (CI) ، وما إلى ذلك بسرعة كبيرة من حيث التبني، تعمل هذه الممارسات على رفع جودة الكود وإمكانية صيانتها، يعد تنفيذ مبادئ الكود النظيف مهارة أساسية تؤتي ثمارها بشكل خاص عندما يحين وقت إعادة بناء الكود، أو وضع الكود قيد الاختبار، حيث يسهل الكود النظيف القراءة والاختبار، إذا كنت تفك في الأمر كجزء من منزل، فإن الكود النظيف هو أساس المنزل.

# أسباب أهمية الكود النظيف

## 3. المنطق الكامن وراء الكود:

إذا سألك شخص ما عن جودة الكود الخاص بك، فيجب عليك تقديم تبرير منطقي، إذا لم تفكر مطلقاً في جودة أسلوب كتابة الكود بشكل منهجي، فمن المحتمل أن يكون هناك الكثير من الفرص للتحسين، أولئك الذين يكتبون كوداً نظيفاً لديهم أنشطة وأنماط وتقنيات ملموسة يستخدمونها لحفظ على الكود الخاص بهم نظيفاً.

## 4. الصيانة:

كتابة الكود سهلة نسبياً، لكن قراءته صعبة، هذا هو السبب في أن العديد من المطورين يفضلون إعادة كتابة الكود بدلاً من القيام بالعمل الشاق المتمثل في قراءة وفهم الكود الموجود، من خلال كتابة الكود يمكن قراءته، فأنت تقوم بتحسين (90%) من الوقت الذي نقرأ فيه الكود، بدلاً من (10%) من الوقت الذي تكتبه فيه، هذه إستراتيجية أكثر فعالية من حيث التكلفة بشكل ملحوظ من الإستراتيجية البديلة لكتابة الكود بأسرع ما يمكن دون القلق بشأن سهولة قراءة الكود.

# أسباب أهمية الكود النظيف

## 5. سهل الاختبار:

من خلال إنشاء كود نظيف، يتم المساعدة على الاختبار الآلي لذلك الكود، من خلال الاختبار الآلي أي الاختبار المستند إلى التطوير، وهو الطريقة الأكثر فاعلية لتحسين جودة الكود، وتحسين سرعة الفريق على المدى الطويل، وتقليل عدد عيوب البرامج، تساهم كل هذه العوامل بشكل كبير في إجمالي عائد الاستثمار للبرنامج.

## 6. البساطة:

حافظ على الكود الخاص بك بسيطًا وقابلًا للقراءة قدر الإمكان، ولا تفرط في تعقيد المشكلات (وهي مشكلة شائعة بين مطوري البرامج)، من خلال إبقائها بسيطة، يمكنك إنتاج كود عالي الجودة وحل المشكلات بشكل أسرع والعمل بشكل أفضل في مجموعات.

## أسباب أهمية الكود النظيف

### 7. الاتساق:

تخيل أنك تذهب إلى متجر ولا يوجد اتساق حول كيفية وضع السلع فيه، سيكون من الصعب العثور على المنتجات التي تبحث عنها، بادئة المسافة في الكود تشبه إلى حد كبير الترتيب الذي تحتاجه في السوبر ماركت، عندما يتم وضع مسافة بادئة لковدك، تصبح أكثر قابلية للقراءة وأسهل للعثور على ما تبحث عنه، خاصة عندما تنتبه إلى أسماء العناصر.

### 8. التوفير في التكاليف:

من خلال كتابة نظيف، ستحصل على كل تلك المزايا المذكورة أعلاه، وكلها تؤدي إلى توفير التكاليف.

شكراً لحضوركم

---

أمل ان تكونوا قد حققتم الفائدة