

البرمجة واللغات البرمجية (هـ)

● **ليست** (ليست بالإنجليزية: LISP أو Lisp لغة تعد إحدى لغات برمجة الكمبيوتر ذات التاريخ العريق، فقد تم وضع مواصفاتها عام ١٩٥٨ وذلك تارخ بعد فورتران Fortran التي طورت قبلها بسنة كإحدى لغة عالية المستوى High-level Pro-Language programming وضعت ليست كلفة ترميز رياضية عملية وفق تعريف تفاضل لامبدا وتكاملها لالوزن تشرش Alonzo Church's Lamb-da Calculus، لكنه سرعان ما فشل استخدامها في أبحاث الذكاء الاصطناعي Artificial Intelligence، ويصنصدها كأحدى أقدم اللغات، قدمت ليست مبادئ عديدة في علوم الحاسب Computer Science كبنية البيانات الشجرية Tree Data Structures Object-oriented Programming.

تشير ليست إلى المصطلح List Processing language، القوائم المتصلة بالإنجليزية: Linked Lists إحدى بنية البيانات الأساسية للغة، بل إن كود المصدر للغة مكون من قوائم، وتتبعه لذلك، تعامل برامج ليست ككود المصدر كبنية بيانات Data Structure يعطي شأنا لنظام الماكرو Macro list ١ 2 foo) يسمح للمبرمجين بإنشاء صيغ جديدة أو لغة مدمجة مخصصة المجال في ليست-مستند Programming Domain-specific Language، ترميز للبيانات ويعطي للغة ليست صيغة تعرف فورية Instantly Recognizable Syntax، وبرامج ليست مكتوبة بشكل التعديل الرمزي S-expression أو إلى Symbol و كقوائم حطاطة بأقواس، فعند استمعا، والة "Function Arguments x وArgs، وازي، تكتب (f x y z)

تاريخ ليست قام باختراع ليست الأمريكي جون مكارثي John McCarthy لعام ١٩٥٨ في معهد ماساتشوستس للتقنية Massachusetts Institute of Technology MIT.

مكارثي نشر تصميمه على الورق في مجلة ACM's Communications of the ACM، وبحسبانها بالة الجزء الأول "Recursive Functions and Their Computation and Symbolic Expressions and the Machine, Part I (Simple Opera-تظهر أنه بواسطة بعض المعاملات البسيطة -Notation، يمكن بنا، لغة تطابق فكرة الشمولية لتورنج Turing-complete لكن من أجل الخوارزميات أول من قام بتطبيق هذا التصميم كان ستيف رسل Steve Russell على جهاز IBM 704، بينما ظهر أول مترجم Compiler ليست كامل كان على يدي تيم هارت Tim Hart ومايك ليفن Mike Levin في معهد MIT عام ١٩٦٢.

اللغة التي قاما بإنشائها أقرب للغة المنتشرة حالياً من صيغتها مكارثي **بعض إصدارات ليست** Arc، Cljure، Standard LISP، Common LISP، إحدى الإصدارات الأكثر انتشاراً.

● **Scheme** إحدى الإصدارات الأكثر انتشاراً، من تطوير غاي إل ستيل وجيرالد جاي سيمان. Arc، Cljure، AutoCAD، الصيغة الإصدار ليست المستخدمة في برنامج Autolisp.

تعد لغة ليست لغة تعبيرة النتيجة Expression-oriented Language، وبخلاف أغلب اللغات، لا فارق بين التعبيرات Expressions والجملة Statements، فالكود يكتب جميعه كمتعبيرات.

لعل ما يميز صيغة كود ليست الأقواس المستخدمة في الإحاطة بين التعبيرات، وقد سبق ذكر المصطلح S-expression الذي يعطي لصيغة ليست استخدام الرموز.

قائمة ليست LISP List يكتب بين قوسين بداخلهما تسرد العناصر مفصولة بمسافة بيشاء، مثلاً:

```
(foo ٢ ١)
```

هذه قائمة بها عناصر تسمى ذرات Atoms، وفي العديدين ١ و٢، العنصر foo من البيانات في ليست يدعى رمز "Symbol"، يتم التعرف على نوع العنصر دون الحاجة للإعلان عنه. القائمة الخالية () تعتبر ذرة خاصة nil بحيث يمكن اعتبارها ذرة إضافية لكونها فائتة.

قد تدخل القوائم كعناصر داخل قائمة ما، مثلاً:

```
((١ ٢ ٣))
```

القائمة السابقة مكونة من عديدين وقائمة بها عديدين. التعبيرات في ليست تكتب كقوائم باستخدام صيغة الرموز أو الـ Prefix Notation. العنصر الأول هو اسم النموذج Form

مثلاً: (Function، معامل حسابي Operator، ماركو Macro، أو معامل خاص Special Operator رسديتي شرحه)، بينما بقية العناصر تعد وسائط Arguments، على سبيل المثال، الدالة list لتعدد وسائطها كقائمة، والتعبير التالي

```
(list ٢ ١ foo)
```

يمثل هذه القائمة (٢ ١ foo)، علامات التنصيص التي تسبق الوسائط تعد إحدى المعاملات الخاصة Special Operators، تمنع علامات التنصيص الوسائط من إجراء الحساب عليها (ليس ذلك ضرورياً مع الأعداد طالما أن العدد ١ هو ١ على المثال، بينما الوسائط التي تحلو من تلك المعاملات فيتم تنفيذها بشكل دوري Recursively قبل الانتهاء من التعبير، امثال التالي:

```
((١ 2 3 4) list ١ 2)
```

يمثل هذه القائمة (٢ ١ ٢ ٣ ٤)، لاحظ أن الوسيط الثالث هو قائمة، فالقوائم يمكن أن تتداخل كما سبق ذكره.

وبالمثل تعامل المعاملات الحسائية، ففي التعبير التالي:

```
((٢ ٢ ١) +)
```

سيتم حساب القائمة وإعادة الناتج ١٠، يمكن توضيح المعادلة نفسها بصيغة الرموز بالداخل "Infix Notation فتكون

```
١+٢+٢+١: المعاملات الحسائية في ليست من نوع ary-ary قابلة لاستقبال أي عدد من الوسائط.
```

الذرات Atoms

في تصميم ليست الأصلي، كان هناك نوعان أساسيان فقط من أنواع البيانات: الذرات Atoms والقوائم Lists. كانت القائمة

سلسلة من العناصر، حيث تعبر كل عنصر ذرة أو قائمة أخرى متداخلة، والذرة قد تكون عدداً Number ورمزاً Symbol، أما الرمز فقد كان عنصراً مميزاً مكوناً من سلسلة من الأحرف والأرقام، وكان يستخدم كاسم متغير أو عنصر بيانات في معالجة الرموز، على سبيل المثال، القائمة (١ 2) BAR) تحتوي

على ثلاث عناصر، الرمز (١ 2) BAR) القائمة (١ 2) BAR) تحتوي

على ثلاث عناصر، الرمز (١ 2) BAR) القائمة (١ 2) BAR) تحتوي

على ثلاث عناصر، الرمز (١ 2) BAR) القائمة (١ 2) BAR) تحتوي

على ثلاث عناصر، الرمز (١ 2) BAR) القائمة (١ 2) BAR) تحتوي

على ثلاث عناصر، الرمز (١ 2) BAR) القائمة (١ 2) BAR) تحتوي

على ثلاث عناصر، الرمز (١ 2) BAR) القائمة (١ 2) BAR) تحتوي

على ثلاث عناصر، الرمز (١ 2) BAR) القائمة (١ 2) BAR) تحتوي

على ثلاث عناصر، الرمز (١ 2) BAR) القائمة (١ 2) BAR) تحتوي

على ثلاث عناصر، الرمز (١ 2) BAR) القائمة (١ 2) BAR) تحتوي

على ثلاث عناصر، الرمز (١ 2) BAR) القائمة (١ 2) BAR) تحتوي

على ثلاث عناصر، الرمز (١ 2) BAR) القائمة (١ 2) BAR) تحتوي

على ثلاث عناصر، الرمز (١ 2) BAR) القائمة (١ 2) BAR) تحتوي

على ثلاث عناصر، الرمز (١ 2) BAR) القائمة (١ 2) BAR) تحتوي

على ثلاث عناصر، الرمز (١ 2) BAR) القائمة (١ 2) BAR) تحتوي

على ثلاث عناصر، الرمز (١ 2) BAR) القائمة (١ 2) BAR) تحتوي

على ثلاث عناصر، الرمز (١ 2) BAR) القائمة (١ 2) BAR) تحتوي

على ثلاث عناصر، الرمز (١ 2) BAR) القائمة (١ 2) BAR) تحتوي

على ثلاث عناصر، الرمز (١ 2) BAR) القائمة (١ 2) BAR) تحتوي

على ثلاث عناصر، الرمز (١ 2) BAR) القائمة (١ 2) BAR) تحتوي

على ثلاث عناصر، الرمز (١ 2) BAR) القائمة (١ 2) BAR) تحتوي

على ثلاث عناصر، الرمز (١ 2) BAR) القائمة (١ 2) BAR) تحتوي

على ثلاث عناصر، الرمز (١ 2) BAR) القائمة (١ 2) BAR) تحتوي

على ثلاث عناصر، الرمز (١ 2) BAR) القائمة (١ 2) BAR) تحتوي

على ثلاث عناصر، الرمز (١ 2) BAR) القائمة (١ 2) BAR) تحتوي

على ثلاث عناصر، الرمز (١ 2) BAR) القائمة (١ 2) BAR) تحتوي

على ثلاث عناصر، الرمز (١ 2) BAR) القائمة (١ 2) BAR) تحتوي

على ثلاث عناصر، الرمز (١ 2) BAR) القائمة (١ 2) BAR) تحتوي

على ثلاث عناصر، الرمز (١ 2) BAR) القائمة (١ 2) BAR) تحتوي

على ثلاث عناصر، الرمز (١ 2) BAR) القائمة (١ 2) BAR) تحتوي

على ثلاث عناصر، الرمز (١ 2) BAR) القائمة (١ 2) BAR) تحتوي

على ثلاث عناصر، الرمز (١ 2) BAR) القائمة (١ 2) BAR) تحتوي

على ثلاث عناصر، الرمز (١ 2) BAR) القائمة (١ 2) BAR) تحتوي

على ثلاث عناصر، الرمز (١ 2) BAR) القائمة (١ 2) BAR) تحتوي

على ثلاث عناصر، الرمز (١ 2) BAR) القائمة (١ 2) BAR) تحتوي

على ثلاث عناصر، الرمز (١ 2) BAR) القائمة (١ 2) BAR) تحتوي

على ثلاث عناصر، الرمز (١ 2) BAR) القائمة (١ 2) BAR) تحتوي

على ثلاث عناصر، الرمز (١ 2) BAR) القائمة (١ 2) BAR) تحتوي

على ثلاث عناصر، الرمز (١ 2) BAR) القائمة (١ 2) BAR) تحتوي

على ثلاث عناصر، الرمز (١ 2) BAR) القائمة (١ 2) BAR) تحتوي

على ثلاث عناصر، الرمز (١ 2) BAR) القائمة (١ 2) BAR) تحتوي

على ثلاث عناصر، الرمز (١ 2) BAR) القائمة (١ 2) BAR) تحتوي

على ثلاث عناصر، الرمز (١ 2) BAR) القائمة (١ 2) BAR) تحتوي

على ثلاث عناصر، الرمز (١ 2) BAR) القائمة (١ 2) BAR) تحتوي

على ثلاث عناصر، الرمز (١ 2) BAR) القائمة (١ 2) BAR) تحتوي

على ثلاث عناصر، الرمز (١ 2) BAR) القائمة (١ 2) BAR) تحتوي

على ثلاث عناصر، الرمز (١ 2) BAR) القائمة (١ 2) BAR) تحتوي

على ثلاث عناصر، الرمز (١ 2) BAR) القائمة (١ 2) BAR) تحتوي

على ثلاث عناصر، الرمز (١ 2) BAR) القائمة (١ 2) BAR) تحتوي

على ثلاث عناصر، الرمز (١ 2) BAR) القائمة (١ 2) BAR) تحتوي

على ثلاث عناصر، الرمز (١ 2) BAR) القائمة (١ 2) BAR) تحتوي

على ثلاث عناصر، الرمز (١ 2) BAR) القائمة (١ 2) BAR) تحتوي

على ثلاث عناصر، الرمز (١ 2) BAR) القائمة (١ 2) BAR) تحتوي

على ثلاث عناصر، الرمز (١ 2) BAR) القائمة (١ 2) BAR) تحتوي

على ثلاث عناصر، الرمز (١ 2) BAR) القائمة (١ 2) BAR) تحتوي

على ثلاث عناصر، الرمز (١ 2) BAR) القائمة (١ 2) BAR) تحتوي

على ثلاث عناصر، الرمز (١ 2) BAR) القائمة (١ 2) BAR) تحتوي

على ثلاث عناصر، الرمز (١ 2) BAR) القائمة (١ 2) BAR) تحتوي

على ثلاث عناصر، الرمز (١ 2) BAR) القائمة (١ 2) BAR) تحتوي

على ثلاث عناصر، الرمز (١ 2) BAR) القائمة (١ 2) BAR) تحتوي

على ثلاث عناصر، الرمز (١ 2) BAR) القائمة (١ 2) BAR) تحتوي

على ثلاث عناصر، الرمز (١ 2) BAR) القائمة (١ 2) BAR) تحتوي

على ثلاث عناصر، الرمز (١ 2) BAR) القائمة (١ 2) BAR) تحتوي

على ثلاث عناصر، الرمز (١ 2) BAR) القائمة (١ 2) BAR) تحتوي

على ثلاث عناصر، الرمز (١ 2) BAR) القائمة (١ 2) BAR) تحتوي

على ثلاث عناصر، الرمز (١ 2) BAR) القائمة (١ 2) BAR) تحتوي

على ثلاث عناصر، الرمز (١ 2) BAR) القائمة (١ 2) BAR) تحتوي

على ثلاث عناصر، الرمز (١ 2) BAR) القائمة (١ 2) BAR) تحتوي

على ثلاث عناصر، الرمز (١ 2) BAR) القائمة (١ 2) BAR) تحتوي

على ثلاث عناصر، الرمز (١ 2) BAR) القائمة (١ 2) BAR) تحتوي

على ثلاث عناصر، الرمز (١ 2) BAR) القائمة (١ 2) BAR) تحتوي

فيما لو وجدت خلية معطاة بمقدمة قائمة متصلة، فالجزء car

يها يحدد العنصر الأول من القائمة، والجزء cdr، ويؤشر إلى باقي

القائمة. لهذا فإن دوال cdr تستدعي أيضا cdr first

الحدث عن خلايا في بنية القوائم المتصلة (بدلاً من البنية الأخرى

كشجرة tree مثلاً). إذا القائمة في ليست لا تعتبر وحدة

أساس، كحال أي نسخة Instance من صنف Class لغة

كجافا أو سي++، التغير الذي يشير إلى قائمة معطاة أو ببساطة

مؤشر إلى الخلية الأولى لتلك القائمة.

ولأن استخدام الخلايا والقوائم شائع بكثرة في أنظمة ليست،

فهناك اعتقاد خاطئ شائع بأنها البنية الوحيدة للبيانات في ليست،

لكن بالواقع، هناك بني أخرى أسهل كوتبنا كالتجهات Hash Tables،

(المصفوفات Arrays)، الجداول المتشابهة Hash Tables.

البينى Structures ومكافئها

تمثيل القوائم المتصلة بالتعبير الرمزية

يمكن تمثيل بنية القائمة المتصلة بأقواس التعبيرات الرمزية بطرق

مختلفة، يمكن كتابة الخلية في القائمة المتصلة بطريقة ترميز

النقطة-الزوج dotted pair notation، مثلاً: (a b)، حيث a

هو الجزء، gear هو الجزء cdr، يمكن كتابة قائمة تامة طويلة

بتلك الطريقة كالتالي:

```
(d (c (b a)))
```

ويمكن اختصار كتابة القائمة السابقة بطريقة ترميز القائمة

List notation كالتالي:

```
(a b c d)
```

يتم كتابة قائمة غير تامة Improper List عند الجمع بين

اثنين (مثلاً مع القائمة بها ثلاثة خلايا كالتالي: (a b c d)،

حيث أن العنصر آخر جزء cdr، القائمة التالية هي الصورة

الصحيحة:

```
((c (b a)))
```

عمليات المعالجة في القوائم

يقدم ليست إجراءات مبنية داخليا Built-in Procedures

للوصول لليس Acces والتحكم بالقوائم Controlling List، يمكن

إنشاء القوائم مباشرة بالإجراء list، الذي يأخذ أي عدد من

الوسائط، ويعيدها كعناصر بالقائمة:

```
(list 1 2 'a 3)
```

مثال آخر،

```
(list 1 (2 3) 4)
```

Output: (1 2 3 4); ولأن القوائم يمكن أن يتم إنشاؤها بشكل أزواج وخلايا،

فالإجراء cons يمكن أن يدرج عنصراً بمقدمة القائمة، لاحظ أن

هذا الإجراء مختلف في تعامله مع الوسائط بسبب الاختلاف في

طرق إنشاء القوائم:

```
(cons 1 (2 3))
```

Output: (1 2 3); مثال آخر،

```
(cons (1 2) (3 4))
```

Output: (1 2 3 4); الإجراء append يمكن أن أكثر من القوائم إلى قائمة

معبئة:

```
(append (1 2) (3 4))
```

Output: (1 2 3 4); مثال آخر،

```
(append (1 2 3) (4 5 6))
```

Output: (1 2 3 4 5 6); البنية الشجرية

بما أن القوائم في ليست قوائم متصلة بسيطة، فهي تستطيع

مشاركة بنية ما مع قوائم أخرى، فيمكن لقائمتين أن تشتركا بذيل

Tail واحد، أو بأخر سلسلة من الخلايا. مثلاً:

```
self foo (list 'a 'b 'c) (self bar (cons 'x (cdr
```

foo)))

القائمة self foo (b c) والقائمة self bar (a b)

(c) (ذيل) (c) (بذيل في بنية كلا من القائمتين فهو ليس نسخة

مكروة.

يؤدي عمل البنية المشتركة إلى تحسن في الأداء بشكل أفضل

من مجرد نسخ العناصر المشتركة، لكن بالمقابل فإن أي تغيير في

البنية المشتركة يؤثر على القوائم المبنية عليها، مثلاً:

```
(self (third foo) 'goose)
```

عند استدعاء العنصر عني القائمة foo بالعنصر goose،

ستتغير القائمة إلى (a b goose) أي القائمة bar ستتغير

أيضاً إلى (x b goose).

التطبيق التقني Self-evaluate وعلامات التنصيص

يقوم ليست بتطبيق (أو حساب) القوائم التي يدخلها المستخدم

إلى صور أخرى، وعادة ما تكون بسيطة، مثلاً، القائمة (٢+٣)

تتخذ ويعاد الناتج ٥، النماذج الأخرى عادة ما تتخذ وتعيد نفسها،

فالعدد ٥ يعيد العدد ٥.

هناك عبارات يمكن وضعها بعد علامات تنصيص لمنع إجراء

عمليات تنفيذ عليها (كما هو ضروري للرموز والقوائم كما ظهر

تلك بالأعلى السابقة)، وهنا يظهر عمل العامل الخاص quote

أخصاره علامة التنصيص الفريدة. مثلاً، عند إدخال الرمز

foo، سيتم تنفيذ وإعطاء القيمة التي يحملها، أو تظهر رسالة

خطأ لو لم يحمل أي قيمة، لكن لو أزدت الإشارة إلى الرمز حرفياً،

فيجب كتابة (foo quote) اختصاراً by writing foo.

البرامج المكتوبة أثناء بصيغة كومون ليست Common LISP:

```
print "Hello world!"
```

برنامج طباعة العبارة "Hello world!";

```
defun factorial (n)
```

```
(if (<= n 1)
```

```
1
```

```
(* n (factorial (- n 1))))
```

المثال السابق يمكن استخدام ماركو الودار loop بدلاً من

التدوير

```
defun factorial (n)
```

```
(loop for i from 1 to n
```

```
for fac = 1 then (* fac i)
```

```
finally (return fac)))
```

إعداد / م : إبراهيم العريفي

com.Ibrahim_ah219@yahoo

أمن المعلومات (٢)

تحدث المشكلة الأمنية عندما يتم اختراق النظام لديك من خلال أحد المهاجمين أو المتسللين (الهacker) أو الفيروسات أو نوع آخر من أنواع البرامج الخبيثة وأكثر الناس المستهدفين في الاختراقات الأمنية هم الأشخاص الذي يقومون بتصنفح الإنترنت، حيث يتسبب الاختراق في مشاكل مزعجة مثل تعطيل حركة التصفح وانقطاعه على فترات منتظمة. ويمكن أن يتعدى الدخول إلى البيانات وفي أسوأ الأحوال يمكن اختراق المعلومات الشخصية للمستخدم وفي حالة وجود أخطاء برمجة أو إعدادات خاطئة في خادم الويب، فمن الجائز أن تسمح بدخول المستخدمين عن بعد غير المصرح لهم إلى الوثائق السرية المحتوية على معلومات شخصية أو الحصول على معلومات حول الجهاز المضيف للخادم مما يسمح بحدوث اختراق للنظام. كما يمكن لهؤلاء الأشخاص تنفيذ أوامر على جهاز الخادم المضيف مما يمكنهم تعديل النظام وإطلاق هجمات إغراقية مما يؤدي إلى تعطل الجهاز مؤقتاً. كما أن الهجمات الإغراقية DDos مستهدفة إما أو شل حركة مرور البيانات عبر الشبكة، كما أنه من خلال الهجمات الإغراقية الموزعة DDos، فإن العددي يقوم باستخدام عدد من الكمبيوترات التي سيطر عليها للهجوم على كمبيوتر أو كمبيوترات أخرى، ويتم تركيب البرامج الرئيسة للهجمات الإغراقية الموزعة DDos في أحد أجهزة الكمبيوتر مستخدماً حساباً مسروقاً. إن التجسس على بيانات الشبكة وأعراض المعلومات التي تتقلع بين الخادم والمتعرض يمكن أن يصبح اسراً مكملاً إذا تركت الشبكة أو الخوادم مفتوحة وتقاط ضعفها مكشوفة.الهacker الهacker هو الشخص الذي يقوم بإنشاء وتعديل البرمجيات والعتاد الحاسوبي، وقد أصبح هذا المصطلح ذا مغزى سلبي حيث صار يطلق على الشخص الذي يقوم باستغلال النظام من خلال الحصول على دخول غير مصرح به للأنظمة والقيام بعمليات غير مرغوب فيها وغير مشروعة. غير أن هذا المصطلح (هاكر) يمكن أن يطلق على الشخص الذي يستخدم مهاراته لتطوير برمجيات الكمبيوتر وإدارة أنظمة الكمبيوتر وما يتعلق بأمن الكمبيوتر فيوسات الكمبيوتر.فيبروسات الكمبيوتر هي الأكثر شيوعاً من بين مشاكل أمن المعلومات التي يتعرض لها الأشخاص والشركات. وفيبروس الكمبيوتر هو برنامج غير مرغوب فيه ويدخل إلى الجهاز دون إذن ويقوم بإدخال نسخ من نفسه في برامج الكمبيوتر، والفيروس هو أحد البرامج الخبيثة أو المتطفلة والبرامج المتطفلة الأخرى تسمى الديدان أو إحصنة طروادة أو برامج الدعساية أو برامج التجسس يمكن للبرامج الخبيثة أن تكون فقط للإزعاج من خلال التأثير على استخدامات الكمبيوتر وتطبيقه وتسبب في حدوث انقطاعات وأعطال في أوقات منتظمة وتؤثر على البرامج والوثائق المختلفة التي قد يرغب المستخدم في الدخول إليها. أما البرامج الخبيثة الأكثر خطورة فهي أن تصبح مشكلة أمنية من خلال الحصول على معلوماتك الشخصية من رسائلك الإلكترونية والبيانات الأخرى المخزنة في جهازك. أما بالنسبة لبرامج الدعساية وبرامج التجسس فهي مزعجة في الغالب وتؤدي إلى ظهور نوافذ دعائية منقطة على الشاشة. كما أن برامج التجسس تجمع معلوماتك الشخصية وتقدمها إلى جهات أخرى تطلب الحصول عليها لأغراض تجارية يمكنك حماية كمبيوترك وحماية نفسك باستخدام برامج مناسبة لمكافحة البرامج الخبيثة غير المرغوب فيها والتي قد تكون نتائجها مدمرة. اللصوصية (Phishing) يستخدم مصطلح Phishing للتعبير عن سرقة الهوية، وهو عمل إجرامي، حيث يخدع شخص أو شركة للتحايل والغش من خلال إرسال رسالة بريد إلكتروني مدعياً أنه من شركة نظامية ويطلب الحصول من مستلم الرسالة على المعلومات الشخصية لن تفاصيل الحسابات البنكية وكلمات المرور وتفاصيل البطاقة الائتمانية. وتستخدم المعلومات للدخول إلى الحسابات البنكية عبر الإنترنت والدخول إلى مواقع الشركات التي تطلب البيانات الشخصية للدخول إلى الموقع هناك برامج لمكافحة اللصوصية Phishing للكشف عن هوية المرسل الحقيقي، وأفضل وسيلة لحماية الشخص من نشر معلوماته الشخصية لن طلبها هو أن يكون الشخص متيقظاً وحذراً ولديه الوعي الكافي، فلا يوجد هناك أي بنك معروف أو مؤسسة فعلية يطلبون من عملائهم إرسال معلوماتهم الشخصية عبر البريد الإلكتروني.