

P3rL M4G4Z!n3(2)

SG.EDITION

اسم الدالة:-**Abs**

التصنيف:-رياضي

القيم المعادة :-رقمية التعريف

هذه الدالة كما هو واضح من اسمها عمل الدالة الرئيسي هو ان اذا قمت بارسال عدد اليها سالب تقوم بارجاعه ولكن موجب اي انها تعمل الى ارسال الحد المطلق لهذه الدالة

***C0d3**

```
$a=-6546;  
print (abs($a));
```

نلاحظ انه اذا قمنا بتنفيذ هذه الدالة ان الرقم المعاد هو نفس الرقم ولكن من دون علامة الناقص التي تشير الى كونه عدد سالب

وهناك طريقة ثانية في استعمال هذه الدالة لكن النتيجة نفس النتيجة ولكن طريقة الاستعمال تختلف وهي كما يلي

***C0d3**

```
print ("abs (-10) equalto ",abs(-10),"\\n"
```

شوف في هذه الدالة لدينا طريقتان وهما لهم نفس الناتج شوف اي واحدة من هذه الطرق تشوفها سهلة استعمالها

اسم الدالة:- **accept**

التصنيف:- socket

القيم المعادة:- (false), " integer (address of socket),
التعريف

هذه الدالة تعمل على انجاز

low level unix socket call accept

حقيقة انا هذه الدوال لم اعرف معظم افعالها دائما لذا سوف

اقوم بشرح ما اعرفه عن دوال ال

unix system call

اسم الدالة:- **alarm**

التصنيف:- الدوال الداخلية الخاصة بالعمليات الاحادية اوالمعالجة

القيم المعادة:- اعداد صحيحة ثواني لل المنبه السابق

التعريف

هذه الدالة تعمل على اعداد اشارات ال

unix SIGALRM

لكي يتم توليدها في خلال عدد من الثواني وهذا العدد من

الثواني انت تقوم بتحديد

واذا لم تقوم بتحديد عدد الثواني فأن القيمة الموجودة في

المتغير

\$_

سوف يتم استخدامها والقيمة التي تكون في هذا المتغير هي

القيمة الديفلت

و انه من الممكن للبيرل ان توقع مثل هذه الاشارات عن طريق

استدعاء اشارة محددة تعالج الروتين الفرعي

استدعاء الصفر على انه معامل يعمل على الغاء الارم الحالي

اسم الدالة:-atan2

التصنيف:-رياضي

القيم المعادة :-رقمية
التعريف

هذه الدالة تعمل على اعادة ال

arc tangent

وهي تعني

arc

تعني قوس الدائرة ولكن لاتعني شي منحرف او مائل و ميل لذا

ارجو الانتباه

tangent

ظل التمام تعني هذه الكلمة زين هذه الدالة رياضية يعني

وكما في هذا الكود الاتي

*C0d3

```
$a=11;  
$b=12;  
print (atan2($a,$b));
```

اسم الدالة: **bind**

المعاملات:- عددي عناوين شبكة
القيم المعادة:- واحد اذا كان صحيح ' ' اذا كانت القيم خطأ
التعريف
هذه الدالة تعمل على ربط عنوان الشبكة الى السوكت

اسم الدالة: **binmode**

التصنيف:- العمليات الاحادية
القيم المعادة:- واحد اذا كانت صح و غير معرف اذا كان هناك اي
خطأ
التعريف
ملاحظة تجدر الاشارة اليها انه هذه الدالة تستعمل مع ال
دائما open
الانظمة التي تميز بين الملفات التي تكون اما بالنمط النصي اي
(Text)
او الملفات التي تكون ذات النمط الثنائي اي ال
(binary mode)
هذه الدالة تجبر على معالجة الملفات ذات النمط الثنائي
كلنا لاحظنا في الملفات النصية انها تحتوي على ما يسمى ب
end of line characters
like "\n" or "\t" etc
فان هذه العلامات مباشرة تترجم الى جمل ينكس ل
end of line
هذا قلنا موجود في الملفات ذات النمط النصي بينما في
الملفات التي تكون ذات نمط بيناري اي ثنائي لاتملك هذا النوع

من الترجمة الالية لجمل نهاية الاسطر

اسم الدالة:-**caller**

التصنيف:- معاملات احادية(scope)

المعاملات:- تعابير

القيم المعادة:- 1 اذا كانت صحيحة ' ' اذا كانت خطأ او

package, filename, line

التعريف

هذه الدالة تستخدم لفحص مجال استدعاء الروتين الفرعي

eval اذا تم عمل

scalar له على انه

فانه سوف يعيد واحد او يعيد ' ' معتمدا على انه الكود هل تم

استدعاه على انه روتين فرعي

طبعا هذا الكود يمكن استخدام معه ال

require()

eval()

يعني هذه من الممكن استخدامه

اما اذا تم استدعاه على اساس انه مصفوفة اي انه تم استدعاه

array على انه

فانه سوف يعيد

package name = اسم الحزمة

file name = اسم الملف

line = السطر

والان سوف نلاحظ هذا المود تم استدعاء الةتين الفرعي على

انه مصفوفة

*C0d3

```
sub spawn {  
  ( $package, $file, $line)=caller;  
}  
&spawn;  
print "Package = $package, File=$file, line=$line";
```

ونلاحظ انه اعاد لنا القيم المطلوب منه ان يعيدها وهي اسم
الباكيج واسم الملف والسطر

اسم الدالة:- **chdir**

التصنيف:-ملفات

المعاملات:- تعابير

القيم المعادة:-1 اذا كانت صحيحة ' ' اذا كانت خطأ
التعريف

هذه الدالة تعمل على تغيير المجلد الحالي الذي تعمل
تحديد المجلد الذي تريد ان تغيره اذا لم يتم اعطاء معامل فأن
القيمة الافتراضية سوف تكون هي
مجلد الهوم

الان سوف نرى كيف تعمل هذه الدالة كما في هذا الكود

*C0d3

```
chdir ("/etc") ? print "it work" : print "not work";
```


اسم الدالة:- **Chmod**

التصنيف:- ملفات

القيم المعادة:- عددية

التعريف

اولا دالة مثل هذه الدالة تكون غنية وغنية جدا عن التعريف وهي الدالة المشهورة باعطاء التصاريح للملفات بدون اي مقدمات هذا كود عن طريقة عمل هذه الدالة

***C0d3**

```
print "chmod() changed ", chmod(0744,"/tmp/test1.txt","/tmp/test2.txt"),  
files.\n";
```

واظن هذه الدالة لاتحتاج الى شرح لانه جميع المبرمجين حتى وان كانوا قليلي الخبرة بالبيرل لكن تكون لديهم خبرة بهذه الدالة

اسم الدالة:- **chomp**

التصنيف:- معاملات ومتغيرات

القيم المعادة:- عددية

التعريف

هذه الدالة هي دالة مرادفة لدالة ال

chop()

حيث انها تعمل على ازالة الرموز من نهاية الاسطر وهي تعيد عدد الرموز التي تم عمل ريموف عليها وكما في الكود الاتي

*C0d3

```
$a="spawn!\n";  
$b=chomp $a;  
print $a,$b;
```

نلاحظ انه عندما يتم تنفيذ الكود انه الناتج هو طبع قيمة المتغير الاول وهي سباون ولكن لن نحصل على متغير السطر الجديد وبجانب الكلمة يوجد الرقم واحد والرقم هذه يدل على انه الرمز الذي تم الفأوه هو رمز واحد ولذا يوجد الرقم واحد بينما اذا نفذنا هذا الكود في غنى عن هذه الدالة سوف نحصل على

*C0d3

```
$a="spawn!\n";  
print $a;
```

الان عند تنفيذ هذان الكودان من الممكن ان تعرف ماهو الفرق بين هاتين الدالتين

اسم الدالة:- **chop**

التصنيف:- سترينك سلاسل نصية

المعاملات:- قوائم و متغيرات

التعريف

هذه الدالة تعمل على ازالة الرمز الاخير من السلسلة النصية

وتعيد ذلك الرمز وهذا يحدث كما في الكود الاتي

***C0d3**

```
$a="spawn";  
$b= chop $a;  
print $a, "\n";  
print $b, "\n"
```

حيث انه اذا قمنا بتنفيذ هذا الكود سوف ينتج من المتغير الاول قيمة المتغير وهي

***C0d3**

```
spaw
```

ونلاحظ انه قد قام بالغاء الحرف الاخير من السلسلة النصية الموجودة ونتاج عنها الكلمة الموجودة في الاعلى والنتاج من تنفيذ المتغير الثاني هو الرمز

***C0d3**

```
n
```

هنا نلاحظ انه قد قام باعادة الحرف الاخير من السلسلة النصية الموجودة في المتغير الاول

اسم الدالة: **chown**

التصنيف: -ملفات

المعاملات: -ملفات

التعريف

هذه الدالة تعمل على تغيير الملكية لمجلد معين موجود في

مكان ما كما في المثال الاتي

*C0d3

```
chown (0,0,"/tmp/vb/bn")? print("Worked\n") : print("Didn't work\n");
```

نلاحظ هنا انه في هذا المثال الموجود في الاعلى انه استعملنا في البداية اسم الدالة من اجل تغيير الملكية ونلاحظ انه في بداية الكود يوجد الرقمان صفر وصفر وهي تشير الى uid and gid

الخاصة للمستخدم الروت يعني انه هذا الملف الموجود مساره في العلامة الاقتباسية سيكون خاص في الروت واذا نجح الكود سيقول لك العملية تعمل ناجحة و الا بالعكس

اسم الدالة:-chr

التصنيف:-سلاسل نصية

المعاملات:-اعداد

القيم المعادة:-رموز (كاركتر)

التعريف

هذه الدالة تعيد الرمز المشار اليه برقم

اسم الدالة: **close**

التصنيف: -ملفات

المعاملات: -معاملة يدوية

القيم المعادة: - 1 اذا كان صحيح ' ' اذا كان خطأ

التعريف

هذه الدالة تعمل على غلق الملف المفتوح اذا كان الملف

pipe المعطى هو يشير الى ملف

فأن البيرل سوف تنتظر الى ان عملية ال

pipng

تكمل

طبعا من المفضل ان تستعمل مع ههذ الدالة اي دالة الاغلاق

ان تستعمل دالة الفتح

open

حتى تتمكن من غلق الملف المفتوح

كود وشرح على هذه الدالة

***C0d3**

```
open (INF, "/spawn/dd.txt");  
$a=close (INF);  
print("close() Returned ", $a, "on success\n");
```

هنا لدينا هذا المثال الذ استعملنا معه في البداية دالة ال او بن

حيث قمنا بفتح هذا المسار ثم احللنا متغير الى الى دالة الغلق

close وهي دالة ال

كما هو واضح ثم قمنا بالطباعة والتي اذا تمت عملية الطباعة

بشكل ناجح سوف نحصل على الجملة التي تدل على انه الكود

ناجح وتنفيذ صحيح

اسم الدالة: **continue**

التصنيف: flow control

التعريف

syntax structure دالة الاستمرار تكون عبارة عن ال attach والذي يسمح بان يتم عمل مع بلوك اخر اي جملة ضمن بلوك دالة الاستمرار يتم evaluate عمل لها قبل ان يعاد البلوك مثال شرح

***C0d3**

```
$i=0;
print "continue() ";
while ($i<10) {
    if ($i % 2)
        { print "${i}o "; next; }
    else
        { print "${i}e ";}
    } continue {$i++}
print "\n"
```

شرح هذا الكود

اولا عبارة عن متغير قيمته تساوي صفر
ثانيا جملة الطباعة تعمل على طبع سلسلة نصية
while ثالثا جملة ال
\$i تقول اذا كانت اذا كانت قيمة المتغير
اصغر من عشرة ثم بعد ذلك نقوم بفتح ال
curly brackets

while اي الاقواس المتعرجة الخاصة بجمل التحكم وجمل ال
منها
\$i ثم بعد ذلك اذا كانت قيمة المتغير
واذا كان هذا المتغير يقبل القسمة على العدد 2 اذا كان يقبل
اذن في هذه الحالة قم بطباعة
\$i المتغير
ثم استعمل دالة التالي
واغلاق البلوك
else ثم بعد ذلك نستعمل ال
i ايضا تطبع المتغير
ولكن بدون استعمال دالة التالي
ثم ايضا اغلاق البلوك والان يحين وقت استعمال دالة التي الكود
i الذي كتبناه هو عنها حيث نقول استمر مع المتغير
and then newline and finish

اسم الدالة:- **crypt**

التصنيف:- قائمة عمليات

المعاملات:- سلاسل نصية

القيم المعادة:- سلاسل نصية

التعريف

هذه الدالة تعمل على تشفير سلسلة نصية المعامل الاول
اذا تلاحظ انه اذا ردت انه تشفير سلسلة نصية من من متغير
واحد تلاحظ انه راح تحصل على خلل من هذا الخل تعرف انه

عدد المتغيرات التي قمت بادخالها الى البرنامج الذي قمت بكتابته غفي لغة البيرل ليس كافي لذا سوف تقوم بادخال متغير جديد ومهما كان هذا المتغير الجديد فانه اول حرفين منه سوف يدخلون في عملية التشفير وكما يلي

*C0d3

```
$a="spawn";  
print (crypt($a));
```

هنا التشفير في حالة المتغير الواحد والذي راح نحصل منه على هذه الاغلاط

*C0d3

```
Not enough arguments for crypt at - line 2, near "$a)"  
Execution of - aborted due to compilation er
```

نلاحظ هنا البرنامج وقد قمنا بادخال متغير واحد اما الان سوف نقوك بادخال متغيران الى البرنامج ونلاحظ الية التشفير كيف سوف تكون

*C0d3

```
$a="spawn";  
$b="perl curse";  
print (crypt($a,$b));  
print "\n";
```

نلاحظ انه اذا نفذنا هذا البرنامج شوف نحصل على سلسلة نصية مشفرة واذا نلاحظ على السلسلة النصية المشفرة ذاتها نلاحظ انها كما يلي

*C0d3

```
pePeWtLPcv3s2
```

إذا الحرفان الاولان هم اول حرفين من المتغير الثاني
كما ذكرنا قبل قليل
اسم الدالة: **cos**
التصنيف: -عمليات احادية عددية
المعاملات: -تعبير
القيم المعادة: - عددية
التعريف

هذه الدالة من دوال الرياضيات اذ تعمل هذه الدالة على اعادة
قيم الكوساين وهي قيم الجيب تمام في العربي كما يلي

*C0d3

```
print "cos() ",cos(60),"\\n";
```

اسم الدالة: **Defined**
المعاملات: -تعبير
القيم المعادة: - 1 اذا كانت صحيحة و ' ' اذا كانت خاطئة
التعريف

هذه الدالة تعمل على اعادة قيمة من النوع البولي معتمدة
على فيما اذا كانت المعاملات قد تم تعريفها ام لا هنالك فرق
المعرفية وبين قيم null مهم وهو بين القيم ال null
غير معرف null الغير المعرفة بعض الدوال تعمل على اعادة
null من اجل ان تشير الى الخطأ بينما البعض الاخر يعيد قيم
معرفة من اجل ان يشير الى نتيجة خاصة

اذن خلاصة الكلام عن هذه الدالة

هي دالة تعمل على اعادة قيمة من نوع بولياني وهناك فرق مهم بين القيم المعرفة و القيم الغير المعرفة من ال

null

حيث هناك دوال تعيد قمي ال

null من اجل ان تشير الى وجود خطأ ودوال اخرى تعيد قيم ال

من اجل ان تشير الى قيمة معينة

التعريف

*C0d3

```
@iexist = (1,2,3);
print("defined() The array \@iexist ");
defined @iexist ? print("exists.\n") :
print("does not exist.\n")
```

هذا الكود يعطي شرح لهذه الدالة حيث ان هذا الكود مكون من
مما يلي

اولا السطر الاول ه يعباره عن مصفوفة بسيطة مكونة من 3
عناصر

السطر الثاني هو سطر خاص بالطابعة تم تحويل المصفوفة فيه
الى رفرنس

الان نستعمل دالة التعريف حيث نتسعملها مع المصفوفة حيث
اذا كانت المصفوفة معرفة وموجودة فإنه سيطبع جملة تدل
على هذا والا سوف يطبع العكس

اسم الدالة:- **delete**

التصنيف:- هاشات

القيم المعادة :- قيم

المعاملات:- تعابير

التعريف

هذه الدالة تستعمل من اجل مسح عنصر من الهاش والية
المسح تتم عند اعطاء المفتاح العنصر عند ذلك يتم تأشيرته لكي
يتم مسحه
وكما يلي

*C0d3

```
%Hash = (1, One, 2, Two, 3, Three);  
print("delete() Deleted ",delete($Hash{ 1 }),"\\n");
```

الان عند التنفيذ سوف نحصل على القيمة سوف نحصل على
الناتج ولكن اذا اردنا ان نعرف هل العملية تمت بنجاح وان الهاش
قد فقد عنصر منه عندئذ سوف نضيف سطر واحد من اجل التاكيد
انه الهاش قد فقط عنصر

*C0d3

```
%Hash = (1, One, 2, Two, 3, Three);  
print("delete() Deleted ",delete($Hash{ 1 }),"\\n");  
print %Hash;
```

الان عندالتنفيذ هذا الكود سوف نحصل على العنصر الممسوح
ومن ثم راح نحصل على الهاش الذي يخبرنا انه يحوي فقط
عنصرين

اسم الدالة: **die**

التصنيف: -لائحة عمليات

المعاملات: -قوائم

القيم المعادة: -error level

التعريف

هذه الدالة تعمل على إنهاء تنفيذ عمل سكريبت مكتوب في لغة البيرل عندما يتم استدعاؤها حيث انها تطبع الناتج على انه STDERR

which mean

standard error

على كل عمل هذه الدالة مطول جدا لذا ساكتفي في هذا القدر والاعداد القادمة من المجلة سوف اقوم بشرح اعمق عمل هذه الدالة من خلال كود بسيط

***C0d3**

```
die("die() Now we can give an example of  
die()...exiting");
```

اسم الدالة: **do**

التصنيف: -flow

المعاملات: -مقاطع اي بلوكات

المعاملات: -تعايير

المعاملات: -روتينات فرعية

القيم المعادة: -خاصة

التعريف

syntax structure هي ايضا عبارة عن
والتي تسمح بالتنفيذ المتكرر لمقطع من الجمل اي جمل
موجودة في بلوكات
القيمة التي تعاد هي ناتج من اخر جملة في البلوك الذي
يحتوي على الجمل
الان كود يشرح عمل هذه الدالة

***C0d3**

```
$i=1;
print("do ");
$return = do {
    print $i, " ";
    $i++;
} until $i==3;
print("Returned $return\n");
```

الان نعمل افور فيو على هذا الكود في البداية لدينا متغير له له
قيمة وهي 1

الجملة الثانية هي علمية طبع سلسلة نصية بسيطة
do ثم متغير اسمه ريترن اي اعادة يساوي الدالة

\$i ثم نعمل على طباعة المتغير

\$i ثم نزيد قيمة المتغير

نستمر في عملية الزيادة الى ان تكون قيمة المتغير تساوي 3
واذن في هذه الحالة سوف نحصل على نتيجة مكونة من رقمين
هما 1 و 2 وبس لم يكمل لانه قلنا له الى ان تصل الى الرقم 3

اسم الدالة: **each**

التصنيف: -هاشات

المعاملات: -متغيرات

القيم المعادة:- مفتاح ,قيمة
التعريف

هذه الدالة تسمح بأعادة التكرار على العنصر الموجود في
مصفوفة كل مرة يتم عمل

eval

وكانت كل القيم قدتمت اعادتها فأن الدالة سوف تعيد قائمة
null

كود يشرح عمل هذه الدالة

***C0d3**

```
%NumberWord = (1, One, 2, Two, 3, Three);  
print("each() ");  
while (($number,$wordform)=each(%NumberWord)) {  
    print("$number:$wordform ");  
}  
print("\n");
```

الان نعمل اوفر فيو على هذه الدالة

السطر الاول هو عبارة عن مصفوفة لها هذا الاسم

NumberWord

تحتوي على هذه العناصر الموجودة في داخلها ثم جملة
الطباعة التي تقوم بطباعة سلسلة نصية ثم بعد ذلك سطر

while يحتوي على جملة ال

تحتوي بداخلها على متغيرين هما المتغير الاول هو خاص للرقام

1و2و3

المتغير الثاني هو خاص

وهو خاص بالجرء الحرفي اي

ون تو ثري وهكذا

وهكذا نقول انه هذان المتغيران يساويان كل المصفوفة

ثم بعد ذلك جملة الطباعة ثم بعد الانتهاء و التنفيذ بنجاح واطن
اصلا الدالة بسيطة و الشرح جعلها اسهل

اسم الدالة: **exec**

التصنيف: -قوائم عمليات المعالجة

المعاملات: - قوائم

التعريف

هذه الدالة تعمل على تمرير السيطرة من سكربت البيرل الى

امر خارجي في النظام ولهذا لا يوجد اي قيمة معادة من هذه

الدالة

الان نأخذ كود على عمل هذه الدالة

***C0d3**

```
exec ("cat /etc/services);
```

الان لو قمنا بتنفيذ هذه الدالة سوف نحصل على قراءة كاملة

/etc/services لملف ال

هذا هو عمل الدالة الكلي

اسم الدالة: **exists**

التصنيف: - هاشات

المعاملات: - تعابير

القيم المعادة: - 1 اذا كانت صحيحة ' ' اذا كانت خطأ

التعريف

هذه الدالة تعمل على التأكد فيما اذا كان العنصر الذي تبحث

عنه موجود او لا كما في الكود الاتي

*C0d3

```
%test = ( One, 1, Two, 2);  
if (exists $test{One})  
{print "exists() returned success.\n";}  
else {print "exists() returned an error.\n";}
```

طبعا الان راح نحصل على الجملة التي تخبرنا انه الكود صحيح ولكن اذا اردنا ان نتأكد من هذه الدالة العمل الذي تقوم به هو عمل صحيح سوف نعمل تعديل على الكود

*C0d3

```
%test = ( One, 1, Two, 2);  
if (exists $test{six})  
{print "exists() returned success.\n";}  
else {print "exists() returned an error.\n";}
```

الان نحن قليل قد عملنا تلاعب في الكود فقط حولنا مكان كلمة واحد خلينا كلمة ستة واذا قمنا بتنفيذ اتركود سوف نحصل على الجملة التي تخبرنا انه الكود خطأ لذا من خلال التلاعب الصغير قد اصبحتنا نعرف انه عمل هذه الدالة الصحيح من ناحية اختبار وجودية العناصر

اسم الدالة:-exp

التصنيف:-عمليات رياضية

المعاملات:-تعايير

القيم المعادة:-اعداد

التعريف

هذه الدالة هي من الدوال الرياضية التي تعمل على اعادة ال
natural log base (e)

to the power of the expression

طبعا هذه الدالة من الدوال والرياضية مو من الدوال البرمجية لذا
لا اعرف ما هي الخرابيط المسئولة عنها
على كود يشرح عملها بس كود مو انا كاتبه

*C0d3

```
print "exp() e**1 is ",exp(1),"\\n";
```

اسم الدالة:-fileno

التصنيف:-عمليات الادخال و الاخراج

المعاملات:-handle

القيم المعادة:-وصف

التعريف

هذه الدالة تعمل على اعادة الوصف للملف المعطى

اسم الدالة:-fork

التصنيف:-معالجة

المعاملات:-لاتوجد

القيم المعادة:-pid

التعريف

هذه الدالة تدعى

Unix fork()

وهي تعيد ال fork sub process او شي يشبه

process id (pid) of child process

يعني تعيد هوية المعالجة من

child process

الى calling process

ان البرنامج المستدعى يجب ان ينتظر على اي child process

سوف يعمل فورك وذلك تجنبنا لخلق بروسيس طويلة

الان كود يشرح هذه الدالة

***C0d3**

```
$pid = fork;  
# Child only prints this  
if ($pid != 0) {  
    print("fork() Forking a process duplicates o/p: $pid \n");  
}  
waitpid($pid,0);  
# Child exits here  
if ($$ != $origpid) { die; }
```

الان هذا الكود اذا نفذناه راح نحصل على البروسيس اي دي منه وهو للصفر

لكن ماذا سوف يحدث لو غير البروسيس اي دي اذا غيرناه راح نحصل على نتيجة مضاعفة الاولى هي صفر حتى وان كانت

البروسيس اي دي التي دخلناها هي ليست صفر
يعني لو غيرنا في الكود الموجود في الاعلى وغيرنا قيمة
\$pid المتغير
من صفر الى واحد خلى نشوف شو راح يصير

*C0d3

```
$pid = fork;  
# Child only prints this  
if ($pid != 1) {  
    print("fork() Forking a process duplicates o/p: $pid \n");  
}  
waitpid($pid,1);  
# Child exits here  
if ($$ != $origpid) { die; }
```

راح نحصل على بروسيس اي عدد 2 واحدة قيمتها صفر و
الاخرى خاصة بالواحد الصفر الاول هو خاص ب
child process

القيمة الاخرى هي خاصة بال بروسيس اي دي الخاصة
بالواحد

اسم الدالة:- **format**

التصنيف:- قوائم العمليات

المعاملات:- فورمات

التعريف

هذه الدالة تعلن عن

output format specification

هذه الفورمات تستخدم في الربط مع دالة الكتابة من اجل السيطرة مخرجات النصوص والمتغيرات لكي تطابق ما يسمى تركيب الشريحة العامة عندما يتم استدعاء دالة الكتابة فان القيم الفعلية للمتغيرات سوف تستخدم وهذا امفيد في كتابة الحقول البرمجية وكتابة النصوص البرمجية السهلة الان برنامج يوضح عمل هذه الدالة

*C0d3

```
format STDIN =  
format() @>>>>>>> @>>>>>>> @>>>>>>>  
    $t1,    $t2,    $t3  
.  
$t1 = One;  
$t2 = Two;  
$t3 = 3;
```

الان نجبي نحاول نطبق الكود نلاحظ انه الكود لا يطبق لماذا اذا نتفحص الكود نلاحظ انه توجد فقرة فيه ناقصة الاوهي انه نحن لم نستدعي دالة الكتابة وهي الدالة التي ذكرنا انه عندما يتم استدعاها يتم استعمال قيم المتغيرات الفعلية للبرنامج

اذن نعدل في الكود من اجل ان يتنفذ

*C0d3

```
format STDOUT =  
format() @>>>>>> @>>>>>> @>>>>>>  
        $t1,    $t2,    $t3  
  
$t1 = One;  
$t2 = Two;  
$t3 = 3;  
write;
```

نلاحظ الان بعد ما ان تمت عملية استدعاء دالة الكتابة اذا نفذ البرنامج راح نحصل على كم كبير من الاخطاء وهي

Format not terminated at - line 8, at end of line
syntax error at - line 8, at EOF

Execution of - aborted due to compilation error

اذن لماذا كل هذه الاخطاء ونحن استدعينا دالة الكتابة اذن هناك فقرة ناقصة نكتب الكود من جديد

*C0d3

```

format STDOUT =
format() @>>>>>> @>>>>>> @>>>>>>
        $t1,    $t2,    $t3
.
$t1 = One;
$t2 = Two;
$t3 = 3;
write;

```

اه الان اذا قمنا في عملية التنفيذ راح نحصل على ناتج لانه نلاحظ انه قبل ما نعمل احلال في المتغيرات في السطر الذي قبله يوجد علامة النقطة الدوت

اذن هنا مربط الفرس انه هذه النقطة هي التي تستخدم في انهاء الفورمات الان هذا هو الحل النموذجي لدالة الفورمات انا كنت متعمد في انه اكتب الكود في عدة اشكال حتى تتوضح الصورة الاكثر عن عمل الدالة

اسم الدالة:- **formline**

التصنيف:- قوائم عمليات

المعاملات:- صور وقوائم

التعريف

هذه الدالة عادة لا يتم استدعائها بصورة مباشرة حيث انها جزء format كامن من عمل دالة

حيث انها تسمح بعمل تلاعب كامل في عملية الفورمات عن طريق اضافة القيم الى ما يسمى

format accumulator (\$^A)

الان نأخذ كود يوضح عمل هذه الدالة

*Cod3

```
$tmp = formline <<'FINISH', Alpha, Beta, Gamma;
formline() @>>>>> @>>>>> @>>>>>
FINISH
print $^A;
```

الان انا خلّيت الكود الرسمي لهذه الدالة على كل انا اظن انه
انت رسمت فكرة عن عمل هذه الدالة انا مسبقا راح اقلّك انه
الفكرة التي رسمتها عن عمل هذه الدالة هي فكرة خطأ
على كل راح اعطيكم برنامج من برمّجتي حتى تعرفون ما هي
عملية عمل هذه الدالة

*Cod3

```
$a=formline <<'Spawn',SEcuRiTYPuRuS;
formline= @>>>>>>>>>>>>>
Spawn
print $^A;
```

اولا ان المتغير الاول الموجود في البرنامج الذي اسمه
\$tmp
يعني انه الناتج من تنفيذ برنامجك سوف يخزن في مجلد ال
/tmp
في دليل النظام هذا عبارة عن اسم متغير يمكن ان تعمله مثل
a ما تريد يعني انا عامله
ثانيا كلمة
مش كلمة ضرورية في البرنامج يعني لو انت FINISH نهاية
غيرتها ما ارح يتنفذ

لا هي كلمة ما ضرورية وانا بدليتها في اسمي

السطر الثاني

formline كلمة

غير مهمة يعني يمكن تنفيذ البرنامج من دونها

لكن بعدها هذه الشخايط مهمة لانه

لو تركت سهم واحد منها برنامج ما راح يتنفذ او راح يكون تنفيذه ناقص

يعني كما يلي

*Cod3

```
$a=formline <<'Spawn',SEcuRiTYPuRuS;  
@>>>>>>>>  
Spawn  
print $^A;
```

الان نقت من السهم اربعة رموز اذن الان لو تنفذ هذا البرنامج راح تلاحظ انه كلمة سكيورتي كوروس راح تكون ناقصة اربعة رموز وتلاحظون انه البرنامج كتبته من دون كلة فورم لاين

ثالثا مدري رابعا

لاحظوا انه في البرنامج الثاني و الثالث لاتوجد كلمة نهاية او فينش لانه استبدلتها في اسمي وانت ايضا تقدر تحط اسمك مكان اسمي

ثم بعد ذلك الطبع وانتهى

اسم الدالة:- **getc**

التصنيف:- عمليات احادية ادخال واخراج

المعاملات:- ملفات handle

القيم المعادة:- رموز

التعريف

هذه الدالة تعمل على اعادة الرمز الموجود في الملف المعطى في برنامج الدالة واذا كان الملف المعطى لا يحتوى على رموز فإن الدالة سوف تعود بقيم فارغة

على كل هذه الدالة هذا الكود سوف يوضح عملها

*Cod3

```
open INF, "/home/spawn/rr.txt";  
print "getc()", getc(INF);  
print "\n";
```

راح تعطيك الرمز الموجود في السطر الاول وانتهى عمل الدالة

اسم الدالة:-glob

التصنيف:-ملفات

المعاملات:-تعايير

القيم المعادة:-قوائم

التعريف

هذه الدالة تعيد قائمة من الملفات النصية الموجودة في ال

pwd

الذي انت فيه حاليا

على كل عمل هذه الدالة كما في هذا الكود

*Cod3

```
@a=glob("/home/spawn/*.txt");  
print @a[0]
```

على كل انا مسبقا قبل ما اعمل الدالة عملت 3 ملفات نصية

اسمها على التوالي

aa.txt

bb.txt

cc.txt

وبما انه المصفوفة عناصرها معنونة من الصفر اذن المقدار صفر
aa راح ياخذ ال
bb والعنصر رقم واحد راح ياخذ ال
cc والعنصر رقم 2 راح ياخذ

اسم الدالة:- **goto**

التصنيف:- جريان

المعاملات:- تعابير

المعاملات:- اسماء

التعريف

هذه الدالة من الدوال التي من غير الممكن ان يتم استخدامها
لكي تدخل الى داخل المنشأ او ما يسمى بداخل

construct

والذي يحتاج الى تنصيب مثل الروتين الفرعي وال

foreach loops

وانها من الممكن ان يتم استخدامها غالبا في ال

dynamic scope

ايضا انه لاري وول مولف او كاتب البيرل يقول انه انا لا اظن انه
توجد حاجة الى استخدام ال

goto

*Cod3

```
print "goto ";
$count = 1;
TESTGOTO: {
    print $count, " ";
    $label = "TESTGOTO";
    if ($count < 2) {
        $count++;
        goto $label;
    }
    else {
        goto FINISH;}
}
FINISH: print "\n";
```

الان نعمل افور فيو على هذا الكود

اولا لدينا جملة الطبع التي تقوم بطبع جملة ال

goto

ثانيا لدينا المتغير قيمته واحد طبعا اسم المتغير هو اسم فقط يعني ممكن ان تكون انت اي اسم تحبه غير هذا الاسم

ثالثا لدينا مقطع برمجي

رابعا نطبع المتغير الذي حددنا قيمته في البلوك البرمجي وكما قلنا انه اسم المتغير قابل للتغيير وان قيمة المتغير ايضا قابلة للتغيير الخطوة الخامسة هي انه متغير جديد قيمة هذا المتغير

هي البلوك البرمجي

الخطوة السادسة هي انه لو كان المتغير القديم الموجود خارج

البلوك البرمجي قيمته اقل من الرقم 2
افتح قوس لانه جملة ال اذا
هي من جمل التحكم في البيرل
ثم المتغير نفسة اذا كانت اصغر من 2
اذا قم بزيادة مقدارها واحد فقط
ثم نغلق القوس الخاص ببلوك جملة ال اذا
ثم فرضا لو كانت القيمة التي دخلناها في جملة ال اذا غير
صحيحة يعني فرضا لو كان الواحد اكبر 2 اقول فرضا بس
فأن الذي سوف يحدث نستعمل الدالة التي نتكلم عنها
والان عمل ههذ الدالة هي انها تقول لو كانت القيم غير صحيحة
يعني لو كان الواحد اكبر من 2
اذهب الى النهاية ولانلاحظ انه قيمة النهاية هي تطبع سطر
جديد فارغ

اسم الدالة:- **grep**

التصنيف:-قوائم العمليات

المعاملات:-تعايير, قوائم

المعاملات:- قوائم , مقاطع برمجية

التعريف

وشكرا جزيلا له لانه سمح لي باستخدام الشرح الكامل لهذه الدالة

طبعا هذا الشرح الذي ساذكره يشملها بشكل كامل

الشرح

هذا الشرح نسا كما ذكرها الاخ ستورم مان حرفيا بدون تعديل

السلام عليكم ورحمة الله وبركاته

نظرا لانني لسه مبتدا في اللينكس فاعذروني علي بساطة الدرس

grep الدرس بسيط جدا وبيتحدث عن الاداه

اولا تعريف الاداه ديت اداة بتبحث داخل النصوص

شوف الاتي

لو انت عندك سكربت بي اتش بي

وعايز تبحث فيه كله عن الكلمتين

include require

يبقي روح للديريكتروي الي هو فيه واعمل الاتي

:رمز

كود

```
grep -r 'include' *
```

هنا هو حيحيبك كل الملفات فيها الكلمة ديت
طيب نفترض انت عايز تعرف ديت في اي سطر؟؟؟
رمز:

كود

```
grep -r -n 'include' *
```

تمام
طيب نفترض انك عايز تشوف متعرف قبلها متيغرات او لا؟
شوف ياسيدي
رمز:

كود

```
grep -r -B (num) 'include' *
```

معناها رقم السطور الي انت عايز تعمل عنها بحث num كلمة

قبل الكلمة المراد البحث عنها
طيب لو بعدها ???
رمز:

كود
`grep -r -A (num) 'include' *`

طيب لو كل السطور ماعدا الكلمة ديت؟؟
رمز:

كود
`grep -r -v (num) 'include' *`

نشرح بس اخر حاجتين
ديت عشان تبحث في فولدرات وفايلات تانيه غير الي -r اولاً
قدامه بس
لان الجريب مش بيخش في الفولدرات من غير ماتقوله
طيب لو انا عايز فولدرين مثل

اديت عشان نحدد ك الفولدرات
واحد اسمه احمد والثاني اسمه محمد
:رمز

كود
grep -r 'include' ahmed mohamed

طيب لو انت عايز تطبع اسم الملف الي طلعت منه الكلمه
بصفتك بتدور في اكثير من ملف

كود
q grep -H -r 'include *'

طيب لو انت عايزه يتوقف بعد عدد معين من الاسطر المطابقه
حتستخدم ده

كود
-m

كود
grep -m 10 -r 'include *'
طيب لو انت عايز تدور في ديفيس مثلا او سو كيت

حتضيف الاوبشن ده

كود

-D read

وحقيقي معناه انه يعامل السوكيت او الدفيس ده كما لو انه
ملف عادي

كود

grep -D read 'include' *

read skip طيب لو انت مش عايز تقراه حقيقي بدل

كود

في حاجه كمان لو انت بتدور علي كلمه في بدايتها - ممكن
تشكل مشكله مع الجريب حيقولك ده مش اوبشن صحيح

[code]grep -r -e 'include' *

hackobacko تم تعديل الموضوع والى شكر ل

وحيتمك التعديل مره كمان بس شويه صغيرين عشان الوقت
tongue.gif معلى

السلام عليكم ورحمة الله

اسم الدالة:-hex

التصنيف:-قوائم علميات

المعاملات:-تعايير

القيم المعادة:- اعداد

التعريف

هذه الدالة تعمل تقييم للتعبير المعطى على انه سلسلة نصية

hexadecimal

وتعمل على اعادة المساوي لهذا التعبير بقيمة ال

decimal

***Cod3**

```
print("hex() ",hex("ff"), "\n");
```

اسم الدالة:- **int**

التصنيف:- عددي

المعاملات:- تعابير

القيم المعادة:- اعداد

التعريف

هذه الدالة تعمل على اعادة الرقم بشكله الصحيح يعني لو كان يحتوي على كسر عشري فإنها تعمل على الغاء الكسر العشري

كود يوضح

***Cod3**

```
$a=125.25;  
print (int($a));
```

اسم الدالة:- **join**

التصنيف:- قوائم

المعاملات:- قوائم, تعابير

القيم المعادة:- سلاسل نصية

التعريف

هذه الدالة انا على وفق رأيي البسيط انا اعتبرها من الدوال التنسيقية في لغة البيرل حيث لنفرض كان لديك مصفوفة وادرت ان تخلي فراغات بين القيم الموجودة في المصفوفة لكن نسيت او ادرت ان تدخل قيمة او عنصر جديد اليها فأنك من خلال هذه

الدالة يكون بمقدورك ان تعمل هذا
الان خلى ناخذ هذا الكود

*Cod3

```
@spawn=(1,2,3);  
print (join("-",@spawn));
```

الان هذا الكود راح يعمل على ادخال الشارحة بين القيم لكن لو اردت مثلا ان تدخل عنصر جديد الى قيمة المصفوفة فان هذه العملية تتم كما يلي

*Cod3

```
@a=("abc "," def ");  
print (join("ghi",@a));
```

هنا يعمل على ادخال 3 احرف بين عناصر المصفوفة

اسم الدالة:- **keys**

التصنيف:- هاشات

المعاملات:- مصفوفات

القيم المعادة:- قوائم

التعريف

هذه الدالة كما ذكرنا تتعامل مع الهاشات لكن كلنا نعرف انه الهاش كل عنصر فيه مكون من زوج الشئ قرينه على كل انت لو اردت ان ان تطبع المفتاح وليس القيمة عليك ان تستعمل هذه الدالة حيث هي توفر لك هذه الخدمة وهي تستعمل كما

يلي

***Cod3**

```
%hash=(apple =>" Red", Banna =>" Yellow ");
print (keys(%hash));
```

الان لو نفذنا هذا الكود راح نحصل على الناتج لكن الناتج راح يكون ناتج مخربط يعني القيم داخله في بعضها لذا من الممكن ان نعمل عمل من شأنه ان يعمل على تحسين شكل join الدالة وهذا من خلال استعمال دالة ال الان شكل الدالة الجديد بعد التحسين يكون كما يلي

***Cod3**

```
%hash=(Apple => "Red", Banna=> "Yellow");
print (join("-",keys(%hash)));
print "\n";
```

والان يمكن تنفيذ هذه الدالة وعند التنفيذ ونشوف شكل الناتج راح نعرف الفرق ونشوف انه هذه الدالة دالة الربط ساعدت بشكل كبير على تحسين شكل الناتج وصار اكثر قبولاً

اسم الدالة:-lc

التصنيف:-سلاسل نصية

المعاملات:-تعبير

القيم المعادة:- سلاسل نصية

التعريف

هذه الدالة عملها هو انها تعمل على قيمة المتغير المعادة اليها
على شكل حروف صغيرة حيث ان اسم الدالة
lc= lower case
اي حالة الاحرف الصغيرة في الانكليزي
وعمل الدالة يكون كما يلي

*Cod3

```
$a="SPAWN";  
print (lc($a))
```

والان عند تنفيذ الدالة نلاحظ انه يعيد جميع الاحرف ولكن بحالة
الاحرف الصغيرة ولن يعود اي حرف بشكل كبير

اسم الدالة:- **lcfirst**

التصنيف:- عمليات احادية سلاسل نصية

المعاملات:- تعابير

القيم المعادة :- سلاسل نصية

التعريف

هذه الدالة اذا كان المتغير المستعمل معها هو مكون من مزيج
من الحروف من الشكل الكبير و الشكل الصغير عند استعمال
هذه الدالة نلاحظ انه هذه الدالة تعمل على اعادة الحرف الاول
من المتغير على انه حرف صغير كما في هذا الكود

*Cod3

```
$a="SeCuRiTGuRuS";  
print (lcfirst($a));
```

الان عند تنفيذ هذا الكود
ان الحرف الاول كمن المتغير قد اصبح ذو شكل صغير

اسم الدالة:- **length**

التصنيف:- عمليات احادية سلاسل نصية

المعاملات:- تعابير

القيم المعادة:- اعداد

التعريف

هذه الدالة اذا ما تم استعمالها فانها تعمل على اعادة طول
السلسلة النصية التي عمل احلال لها الى المتغير كما في هذا
الكود

***Cod3**

```
$a="SeCuRiTGuRuS";  
print (length($a));
```

الان عند تنفيذ هذه الدالة نلاحظ انه البرنامج يعيد الينا رقم 13
وهذا يعني ان طول هذه الكلمة هو 13 حرف

اسم الدالة:- **listen**

التصنيف:- سوكت
المعاملات:سوكت كيو اي طابور
القيم المعادة:- 1 اذا كان صح ' ' اذا كان خطأ
التعريف
هذه الدالة مساوية لدالة
UNIX listen() system cal
حيث اذا كنت تستخدم ال
accepts
على السوكت فان دالة ال
listen
تخبر النظام انها موجودة

اسم الدالة:-log

التصنيف:- قوائم عمليات رياضية

المعاملات:- تعابير

القيم المعادة:- اعداد

التعريف

هذه الدالة تعيد اللوغاريتم للتعبير المعطى في الكود البرمجي
كما في هذا الكود

***Cod3**

```
print("log() ",log(2.5),"\n");
```

اسم الدالة:-m///

التصنيف:- عمليات احادية وحدات المطابقة

المعاملات:- m/<pattern>/<optionlist>

المعاملات:- /<pattern>/<optionlist>

القيم المعادة:- 1 اذا كان صح ' ' اذا كان خطأ

التعريف

هذه الدالة اولا قبل التعريف بها اذا اردت ان تعرف كيف تعمل
هذه الدالة عليك ان تكون على دراية بمبادئ التعابير
القاسية الموجودة في لغة البيرل لانه هذه الدالة تعمل وفق
هذه المبادئ

هذه الدالة تعمل على البحث في السلسلة النصية وهذا البحث طبعا يكون عن التطابق وهي تعيد واحد اذا كانت التطابق موجود والا فانها سوف تعيد ال ' ' في حالة عدم وجود التطابق الذي تبحث الدالة في البرنامج وطبعا نحن من نحدد ما هو الذي تبحث عنه الدالة والان نأخذ برنامج يوضح عمل الدالة

*Cod3

```
$_ = "Happy MaN";  
print "m// ",/n$/i,"\n"
```

الان في البداية لدينا متغير قيمته هي سلسلة نصية تحتوي على كلمة هبي مان

m/// ثم الخطوة الثانية من البرنامج توجد كلمة ال وهي اسم الدالة

n ثم الحرف

وهو الحرف الذي نريد ان نبحت عنه ثم تليه علامة الدولار \$

التي تعني انه البحث عن حرف يطبق الحرف n

سوف تبدأ من النهاية

ثم بعدين يجي الحرف

i

هنا نلاحظ انه الحرف الذي نريد ان نبحت عنه هو حرف ذو شكل صغير والحرف الموجود في المتغير هو حرف كبير اذن هنا تبرز اهمية الحرف

i

انه هنا اهميته تبرز انه لا توجد اهيمة لحالة الحروف كلها
الكبيرة و الصغيرة تعامل بشكل واحد
ثم بعدين خطوة الطباعة على سطر جديد وكل شي انتهى
اسم الدالة: **map**

التصنيف:-قوائم عمليات
المعاملات:- لوائح وقوائم برمجية
المعاملات:-قوائم ,تعبير
القيم المعادة:- قوائم
التعريف

هذه الدالة تعمل كل مرة على تقييم التعبير المعطى لها وتعمل
على اعادة القيمة ولكن هذا التقييم الذي تعمله هو تقييم لكل
عنصر موجود في المصفوفة وهذا كله يتم عن طريق الاحلال
المصفوفة الى المتغير الديفلت وهو
\$_

الان عمل هذه الدالة يكون كما يلي

*Cod3

```
@a=map($_+1,(1,2,3));  
print @a;
```

الان عندما ننفذ هذا الكود راح نحصل على قيمة 234
اذن لماذا نحصل على هذه القيم ليش ما نحصل على غير قيم
اذن الحل هو انه انت عندما عملت احلال للمتغير الديفلت
نلاحظ انه يوجد بعد المتغير الديفلت مكتوب +1

اذن من هنا جاء الحل نحن كان لدينا 123
قمنا بعملية الزيادة فتم الناتج 234
وهذه كل العملية
يعني المتغير الديفلت \$ _ تم عمل احلال له لكل عنصر موجود
في المصفوفة

اسم الدالة: **mkdir**

التصنيف: -قوائم عمليات ملفات
المعاملات: - اسماء ملفات
القيم المعادة: -1 او صفر
التعريف

هذه الدالة غنية عن التعريف لانه كل شخص يستعمل الكمبيوتر
يجب ان يكون لديه علم عن عمل هذه الدالة حيث انه عمل هذه
الدالة هو انشاء المجلدات وكما في الكود الاتي سوف يتضح
عمل هذه الدالة

***Cod3**

```
print("mkdir() ",mkdir("/tmp/testdir",0777), "\n");
```

الان اذا كان هذا المجلد غير موجود لديك في الحاسبة فأنها
سوف تعيد لك رقم واحد وهذا دلالة انه عمل الدالة صح
ولكن لو كان المجلد الذي قمت بكتابته هو مجلد موجود في
الحاسبة فأن الدالة سوف تعمل على ارجاع الرقم صفر دلالة
على انه يوجد خلل

اسم الدالة:-next

التصنيف:-flow

المعاملات:-label

القيم المعادة:-N/A

التعريف

هذه الدالة تسمح بالتشعب او التفرع داخل داخل اللوب لذا ال
التنفيذ سوف يقفز الى الطلب التالي للفترة
الان كود يشرح عمل هذه الدالة

*Cod3

```
print("next ");  
@array = ("a","b","c");  
foreach $elem (@array) {  
    next if $elem =~ /^a/;  
    print $elem;  
}  
print "\n";
```

الان لدينا هذا الكود الذي يوضح عمل هذه الدالة الان نعمل اوامر
فيو على هذا البرنامج

في البداية جملة الطباعة التي تعمل على طباعة سلسلة
نصية

ثم بعد ذلك لدينا مصفوفة مكونة من 3 عناصر وهي الحروف الثلاثة
الاولى في الانكليزي ثم بعد ذلك نعمل جملة الفور بواسطة
عنصر قمنا بتسميته

\$elem

يعمل لوب على عانصر المصفوفة ثم بعد ذلك الخطوة التي تأتي بعده هي نخبره ان يوقم بطباعة العناصر التي تأتي بعد الحرف

a

\$elem ثم بعد ذلك نطبع المتغير

اسم الدالة: **no**

التصنيف: -قوائم عمليات موديلات

المعاملات: - موديلات

القيم المعادة: -N/A

التعريف

هذه الدالة يتم استعمالها مع الموديلات وخصوصا ما يعرف ب ال

pragmatic modules

وهي العكس تماما من

use

وعمل الدالة هو كما يلي

*Cod3

```
use intger;  
no intger ;
```

اسم الدالة: **oct**

التصنيف: - قوائم عمليات عددية

المعاملات: -تعايير

القيم المعادة:- اعداد

التعريف

هذه الدالة تعمل تقييم التعبير المعطى على انه سلسلة نصية
من نوع

octal

وتعمل على اعادة القيمة بالصورة العشرية كما في هذا الكود

***Cod3**

```
$a=1000;  
print (oct($a));
```

اسم الدالة:- **ord**

التصنيف:- عمليات احادية سلاسل نصية

المعاملات:- تعابير

القيم المعادة :- اعداد

التعريف

هذه الدالة تعمل على اعادة الرقم او التسلسل الخاص

بالرقم المعطى في البرنامج في ال

ASCII code

للمرمز الموجود في البرنامج

كود يوضح العمل

*Cod3

```
$a="A";  
print (ord($a));
```

الان نلاحظ انه عند تنفيذ البرنامج راح نحصل على قيمة الرمز
A

in ASCII code

وكانت القيمة 65

الان نجرب مثال اخر لنرى مل الدالة بشكل اوضح

*Cod3

```
$c="a";  
print (ord($c));
```

الان عند تنفيذ هذا الكود نحصل على رقم اخر غير الرقم الذي
حصلناه في الحالة الاولى اذن حالة الاحرف من حالتها الكبيرة
الى حالتها الصغيرة تختلف لذا يجب الانتباه الى مثل هذه
الحالات

اسم الدالة:- **pipe**

التصنيف:- قوائم عمليات معالجة

المعاملات:- **readhandle, writehandle**

القيم المعادة:- 1 اذا كانت صحيحة ' ' اذا كانت خطأ
التعريف

هذه الدالة تسمى
pipe
وهي تشبه
UNIX function pipe()

اسم الدالة: **pop**
التصنيف: -عمليات احادية مصفوفات
المعاملات: -متغيرات
القيم المعادة: -قيم
التعريف

هذه الدالة تعمل مع المصفوفات على انها المعاملات الخاصة بها
وسنرى ماذا تعمل من خلال هذا الكود

*Cod3

```
@a = (1,2,3,4);  
print (pop(@a));
```

تعمل على ازالة العناصر الاولى من المصفوفة هذا هو عمل
هذه الدالة

اسم الدالة: **pos**
التصنيف: -عمليات احادية وحدات المطابقة
المعاملات: -متغيرات

القيم المعادة:- اعداد
التعريف
offset هذه الدالة تعمل على اعادة ال
pattern الاخير الذي قام ال
بمطابقته
يعني عمل الدالة يبين كما في هذا الكود

*Cod3

```
$a="Spawn1 Spawn2 Spawn3";  
$a =~ m/Spawn/g;  
print (pos($a));
```

الان نلاحظ عند تنفيذ الكود نلاحظ اننا راح نحصل على الرقم 5
اذن ما هو هذا الرقم خلي ننفذ كود اخر حتى تعرف ما هي
هذه فقرة الارقام

*Cod3

```
$a="Securitygurus1 Securitygurus2 Securitygurus3 ";  
$a =~ m/Securitygurus/g;  
print (pos($a));
```

الان عندما نقوم بتنفيذ هذا الكود نلاحظ انه الرقم الذي يعاد الينا
هو الرقم 13 اذن ان عمل هذه الدالة هي انه تعيد الينا ارقام
الباترون الذي يتم استخدامه في عمليات المطابقة

اسم الدالة:- **push**

التصنيف:- قوائم عمليات مصفوفات

المعاملات:- قوائم, مصفوفات

القيم المعادة:- اعداد

هذه الدالة تستعمل مع المصفوفات حيث انك عندما تريد ان تضيف عنصر الى مصفوفة ولكن لم تصيفها عند بناء المصفوفة فان هذه الدالة هي التي تساعدك حيث انها تعمل على ادخال قيم الى المصفوفة والقيم التي تدخلها هي قيم انت الذي تحددها وعند التنفيذ فان الدالة تعمل على اعادة رقم العناصر المضافة

عمل هذه الدالة كما يلي

*Cod3

```
@a=(1);  
$b=push(@a,2,3,4,5);  
print $b;
```

الان عند تنفيذ الدالة نلاحظ انه الرقم الذي يظهر لنا كنتاج لعملية التنفيذ هو رقم خمسة ولكن اذا اردنا ان نعرف في نفس الوقت الارقام التي اضيفت وعدد العناصر التي اصبحت في المصفوفة نعمل ما يلي

*Cod3

```
@a=(1);  
$b=push(@a,2,3,4,5);  
print $b;  
print "\n";  
print @a;
```

الان نلاحظ عند التنفيذ ان الدالة تعيد لنا عناصر المصفوفة

اسم الدالة:- **quotemeta**

التصنيف:- تعابير

القيم المعادة:- سلاسل نصية

التعريف

هذه الدالة تعمل على اعادة قيمة التعابير المستخدمة بها
وجميع ال

metachatacters

بها قد تم عمل لهم باك سلاش كما في المثال

***Cod3**

```
print(quotemeta("quotameta() I can use any metcharacter $ \ "),"\n");
```

اسم الدالة:- **qw/STRING/**

التصنيف:- **qw/string/**

القيم المعادة:- قوائم

التعريف

هذه الدالة تستعمل بالاحرى تعمل على اعادة سلسلة نصية
اي اقصد تعيد سترينك واذا كانت السلسلة النصية المستعملة
مع الدالة قبل التنقيذ تحتوي على فراغات فانه عند التنفيذ لا

يتم اعتبار هذه المتغيرات مثال على عمل الدالة

*Cod3

```
print("qw// ",qw("1 2 3 4 5"),"\n");
```

نلاحظ في الدالة انه قبل التنفيذ كانت الدالة تحتوي على فراغات في البرنامج لكن بعد التنفيذ نلاحظ انه الدالة قامت بعمل اهمال لهذه المتغيرات وان الناتج هو من دون فراغات هو
12345

اسم الدالة:-rand

التصنيف:-عمليات احادية عددية

المعاملات:-تعايير

القيم المعادة:- اعداد

التعريف

هذه الدالة تعمل على اعادة ارقام عشرية او كسرية يعني المبرمجون يطلقون عليها اسم فلوت اي بها جزء عشري على كل هي تعمل على اعادة ارقام عشوائية قد تكون هذه الارقام تساوي الصفر واهيانا يكون اكبر منه

طرق استعمال هذه الدالة تكون كما يلي من خلال هذا الكود

*Cod3

```
print (rand(10));
```

الان عند التنفيذ نلاحظ انه العدد المعاد هو كبير ولكن في نفس الوقت لو نفذت الدالة مرة ثانية فانك راح تحصل على قيمة جديدة

الان لو انت كنت تريد تحصل على هذا الرقم من درون الكسور العشرية فانت في لغة البيرل كما نعرف من الممكن ان نقوم بدمج بين دالتين من اجل نحصل على رقم عشوائي بدون الكسور العشرية وهذا يتم كما يلي

*Cod3

```
print (int(rand(10)));
```

الان عند تنفيذ هذه الدالة راح نلاحظ انه الرقم المعاد رقم صحيح

اسم الدالة:- **read**

التصنيف:- لوائح عمليات ادخال اخراج

المعاملات:- **handle, variable, length, offset**

المعاملات:- **handle, variable, length**

القيم المعادة:- واحد اذا كانت صحيحة ' ' اذا كانت خطأ

التعريف

هذه الدالة تستعمل لقراءة عدد معين من الحروف من ملف انت تحدده عدد الحروف او طول الحروف التي تقرأها انت الذي

تحدده كما في هذا الكود

*Cod3

```
open(INF,"/etc/services");  
read(INF,$result,10);  
print("read() $result \n");  
close(INF)
```

الان شرح لهذا الكود هو انه المف الذي نقرأ الرموز هو الملف
/etc/services

وعدد الحروف التي سوف نقرأها هي عشرة فقط
ويمكن تغير العدد الى عدد اكبر او اصغر حسب رغبة المبرمج

اسم الدالة:- **rename**

التصنيف:-لوائح عمليات ملفات

المعاملات:- اسماء قديمة ,اسماء جديدة

القيم المعادة:- واحد اذا كانت صحيحة 0 صفر اذا كانت فاشلة
التعريف

هذه الدالة كما هو واضح من اسمها انها تعمل على تغيير
اسماء الملفات وعمل هذه الدالة يتم كما يلي

*Cod3

```
$returned = rename("/tmp/test2","/tmp/test3");  
print("rename() returned $returned \n");
```

نلاحظ عند التنفيذ انه الدالة قامت باعادة الرقم واحد اي ان
العملية تمت بنجاح

اسم الدالة:- **return**

التصنيف:-لوائح عمليات

المعاملات:-قوائم

القيم المعادة:-قوائم

التعريف

هذه الدالة تورد من روتين فرعي مع القيمة المخصصة كما في هذا الكود

***Cod3**

```
sub test {  
  return 1; }  
$test = &test;  
print("return() Returned $test \n");
```

اسم الدالة:- **reverse**

التصنيف:-لوائح عمليات

المعاملات:-لوائح

القيم المعادة:-لوائح

التعريف

هذه الدالة كما هو واضح من اسمها تعمل على عكس القيم الموجودة فيها ونوع المتغيرات التي تتعامل معها هي مصفوفات

***Cod3**

```
@a=(1,2,3,4);  
print (reverse(@a));
```

عند التنفيذ هذه الدالة تعمل على عكس ترتيب العناصر الموجودة في المصفوفة

اسم الدالة:- **s///**

التصنيف:- وحدات مطابقة باترون

المعاملات:- **s/pattern/repalcement/option**

القيم المعادة:- قيم

التعريف

هذه الدالة لن اقوم بشرحها لانني قد فعلت ذلك مسبقا مع في العدد الاول من المجلة وفي موضوع التعابير القياسية اذا كنت تريد تتعلم عن عمل هذه الدالة اقرا ذلك الموضوع

اسم الدالة:- **shift**

التصنيف:- عمليات احادية مصفوفات

المعاملات:- مصفوفات

القيم المعادة:- قيم

التعريف

هذه الدالة تعمل على اخذ العنصر الموجود في اقصى اليسار في المصفوفة التي نعمل عليها وبذلك يقل عدد عناصر المصفوفة بمقدار واحد ولكن عندما لا يتم تحديد مصفوفة فإن

المتغيرات الموجودة في المصفوفة سوف يتن تمريرها الى
المتغير المميز
\$ARGV

اسم الدالة:-sin

التصنيف:-معاملات احادية

المعاملات:-تعايير

القيم المعادة:-اعداد

التعريف

هذه الدالة من الدوال الرياضية التي تعمل على تزويد المبرمج
بما يسمى بمفهوم الجيب في علم الرياضيات
وعمل هذه الدالة يكون كما يلي في هذا الكود

***Cod3**

```
print("sin() ",sin(4), "\n");
```

الان عند تنفيذ هذا الكود نلاحظ ان الدالة تعمل على اعادة
الجيب للرقم

اسم الدالة:-sleep

التصنيف:-عمليات احادية معالجة

المعاملات:-تعايير

القيم المعادة:-قيم

التعريف

هذه الدالة كما هو واضح من اسمها تعمل على ادخال مترجم البيرل في حالة نوم ويكون عمل هذه الدالة على نوعين النوع الاول من عمل الدالة هو يكون كما في الكود الاتي

*Cod3

```
sleep
```

الان عند استخدام هذه الدالة بهذه الطريقة فانه يمكن القول انه مترجم البيرل سوف يدخل في غيبوبة الى الابد لذا لاتقوم بادخال هذه الدالة مع برنامج انت تحتاج الى ناتج التنفيذ له بصورة مستعجلة لانه من شان هذه الدالة هو التأخير الاستعمال الثاني لهذه الدالة

هو انه يتم استعمالها مع رقم وهذا الرقم هو ثواني وهذه الثواني هي عدد الثواني التي من خلالها سوف يدخل مترجم البيرل في الغيبوبة يتم استعمال هذه الطريقة من خلال هذا الكود

*Cod3

```
sleep(5);
```

الان هذا الكود تنفيذه سوف يحتاج الى 5 ثواني فقط لكي يتم التنفيذ حيث رقم خمسة تشير الى خمس ثواني من التأخير وانتهى

اسم الدالة:- **sort**

التصنيف:-لوائح عمليات

المعاملات:-لوائح اسماء فرعية

المعاملات:-لوائح برمجية

المعاملات:-لوائح القيم المعادة:-لوائح التعريف

انا ما ادري ليش هذه الدالة من خلال شرحها الرسمي جعلوها
دالة معقدة جدا وصعبة على كل انا راح اشرحكم عمل دالة
وفق منظوري البرمجي البحت

انت ما عليك من هذه الخرابيط الموجودة فوق تحت خانة
المعاملات هذه الدالة تتعامل مع متغيرات من نوع مصفوفات
هذه الدالة تعمل على ترتيب العناصر الموجودة في المصفوفة
الواحدة وفق التسلسل الابددي اي من اول حرف الى اخر حرف
والان راح نشوف ما هو عمل هذه الدالة

*Cod3

```
@a=("a","c","l","k","m","z","q","v");  
print (sort(@a));  
print "\n";
```

الان عند تنفيذ الدالة نلاحظ انه الدالة تعيد لنا لائحة فيها نفس
العناصر الموجودة في المصفوفة لكن هذه العناصر قد تم ترتيبها

اسم الدالة:- splice

التصنيف:-لوائح عمليات مصفوفات

المعاملات:-array,offset,length,list

المعاملات:-array,offset,length

المعاملات:-array,offset

القيم المعادة:-لوائح

التعريف

هذه الدالة نوع المتغيرات التي تتعامل معها هو المصفوفات وتعمل هذه الدالة على عمل تبديل بين نوعين من العناصر النوع الاول يكون من داخل المصفوفة والنوع الثاني من خارج المصفوفة

يعني العناصر التي سوف تبدل سوف تلغي من المصفوفة وتبقى مكانها العناصر الجديدة

*Cod3

```
@a=("s","p","a","w","n");  
print (splice(@a,"A","N"));
```

الان نلاحظ عند تنفيذ هذا الكود نلاحظ انه لا تتم عملية التغيير بين العناصر الموجودة بين داخل المصفوفة والعناصر الموجودة خارج المصفوفة اذن لابد من انه يوجد خلل لانه لو تم تنفيذ البرنامج كان لابد ان يتبدل مكان ال a الصغيرة

كبيرة وكذا الحال مع الحرف الاخير A كان لابد ان يكون اذن الان سوف نعمل تعديل على هذا الكود

*Cod3

```
@a=("s","p","a","w","n");  
print (splice(@a,0,3,"S","A"));
```

الان عند تنفيذ هذه الدالة نلاحظ انه ناتج الذي تعيده الدالة الينا هو الاحرف

spa

اذن ما الذي اختلف عن الحالة السابقة هو انه فقط الرقم في العناصر في المصفوفة الاولى لكي يتم تبديلها مع المعامل الجديد

اسم الدالة:- **sqrt**

التصنيف:- عمليات احادية رياضية

المعاملات:- اعداد

القيم المعادة:- اعداد

التعريف

هذه الدالة ايضا من الداول الرياضية التي تعمل على اعادة الجذر التربيعي كما في هذا المثال

*C0d3

```
$a=4;  
print (sqrt($a));
```

اسم الدالة: **time**

التصنيف: -اوقات

المعاملات: -لاتوجد

القيم المعادة: -وقت

التعريف

هذه الدالة تعمل على عادة الوقت بالثواني اي تعيد الوقت بصيغة الثواني فقط منذ تاريخ 1 يناير عام 1970 صيغة استعمال هذه الدالة يكون كما يلي

***C0d3**

```
print time();
```

اسم الدالة: **uc**

التصنيف: -عمليات احادية سلاسل نصية

المعاملات: -تعايير

القيم المعادة: -سلاسل نصية

التعريف

هذه الدالة تعمل على اعادة الحروف الموجودة في السلسلة النصية جميعها في حالة الحروف الكبيرة حيث انه اسم الدالة هذا يعني

upper case

يعني حتى وان كانت الاحرف الموجودة في السلسلة النصية هي صغيرة فانها كلها سوف تعيد كلها كبيرة كما يلي

***C0d3**

```
$a="spawn";  
print (uc($a))
```

الان نلاحظ انه عند تنفيذ الدالة انه كل الحروف قد اعيدت في الحالة الكبيرة

اسم الدالة:- **ucfirst**

التصنيف:- عمليات احادية سلاسل نصية

المعاملات:- تعابير

القيم المعادة:- سلاسل نصية

التعريف

هذه الدالة تقريبا عملها يشية الدالة التي فوقها لكن الحرف الوحيد الذي تعيده في حالة الحروف الكبيرة هو الحرف الاول من التعبير الذي تم استعماله مع الدالة كما في هذا المثال

*C0d3

```
$a="securitygurus";  
print (ucfirst($a));
```

اسم الدالة:- **values**

التصنيف:- عمليات احادية هاشات

المعاملات:- هاشات

القيم المعادة:- لوائح

التعريف

اذا كنت قرئت هذه المجلة بشكل صحيح فانك راح تعرف انه
هذه الدالة تعمل بالضبط عكس دالة ال

keys

عمل هذه الدالة يكون كما يلي

*C0d3

```
%hash=(Apple => "Red", Banna=> "Yellow");  
print (join("-",values(%hash)));  
print "\n";
```

اظن انه عمل هذه الدالة صح اخذت نفس مثال دالة ال كيز
وبس بدلت اسم الدالة وايضا خليت دالة الربط جوين حتى شكل

الناتج يكون مقبول وغير مخربط

**NoT3*

This document has been created under GPL license u can copy it or edit in anything u want and remove any thing u don`t like

u can update it without an advance premission

but u should say that u take it from [WWW\[dot\]SecurityGuRus\[dot\]nEt](http://WWW[dot]SecurityGuRus[dot]nEt)

Now the magazine is done
but i wanna say Gr33tz to
(HACKERS PAL) for being a good friend
for a long time
(rOCK-MaStEr) for give the support i
need
(Storm_man) for let me take his subject
about **GREP**
and thx to every one in SG,to Be
Really g00d friends

M4ILm3:-

mahmoud_najafy@hotmail.com

M_SpAwN