

انواع ال shell في انظمة التشغيل مع الفرق في كل نظام تشغيل؟

1. في نظام تشغيل Unix/Linux:

- Bash (Bourne-Again SHell): هو Shell هو Linux. Bash الافتراضي في أغلب توزيعات. مع ميزات إضافية (sh) Bourne Shell تطوير ل.
- Zsh (Z Shell): shell أخرى شائعة الاستخدام في Bash وتعتبر أكثر تطوراً من Unix/Linux، في بعض الميزات.
- Fish (Friendly Interactive SHell): shell جديدة نسبياً تهدف إلى توفير تجربة أكثر ودية وسهولة الاستخدام.
- Tcsh (TENEX C Shell): C نسخة محسنة من وهي شائعة الاستخدام في بعض Unix. توزيعات Shell (csh).

2. في نظام تشغيل Windows:

- PowerShell: shell خاصة بنظام التشغيل

Command و تُعتبر البديل الأقوى لـ Windows، Prompt.

- Command Prompt (cmd.exe): shell وهي أبسط وأقل قوة من Windows الافتراضية في PowerShell.

3. في نظام تشغيل macOS:

- Zsh (Z Shell): shell الافتراضية في macOS وأحدث Catalina.

- Bash (Bourne-Again SHell): كان shell قبل macOS الافتراضي في Catalina.

الفروق الرئيسية بين هذه الأنواع من الشل:

- الميزات المتقدمة: Zsh و Fish هي الأكثر تطوراً من حيث ميزات التكملة التلقائية، وإدارة المهام، والتعامل مع أدوات أخرى.

- التوافق والسهولة: Bash و Command Prompt هما الأبسط والأكثر توافقاً مع الأوامر القياسية.

- المرونة والقوة: PowerShell هي الأقوى في التعامل مع الكائنات (Objects) وإدارة النظام.

- الشكل والمظهر: Fish تتميز بواجهة مستخدم رسومية جميلة وأكثر .

ماهو ال regex وماهي الاوامر التي تستخدم فيه؟

هي لغة نمطية Regex (Regular Expressions) تُستخدم لتعريف أنماط نصية معقدة. تتيح لك البحث والتعرف على أنماط محددة في النصوص بشكل دقيق وفعال.

بعض من أهم الأوامر والعناصر الخاصة بالتعبيرات النمطية (Regex):

1. ****الأحرف الأساسية****: تتضمن الأحرف الإنجليزية الصغيرة والكبيرة، والأرقام، والأحرف الخاصة مثل `(\|{}[]() ?+*^$_.`)`.

2. ****الحروف الخاصة****:

- ``\d``: تطابق أي حرف واحد عدا الخط الجديد.

- ``\d``: تطابق أي رقم (0-9).

- ``\w``: تطابق أي حرف أبجدي أو رقم أو السّفْطَرة.

(`_`).

- ``s` (مسافة، TAB، خط جديد، إلخ).
تطابق أي حرف فراغ: ``s` -

3. **تكرار الحروف**:

- ``*`: تطابق الحرف السابق له 0 أو أكثر مرات.
- ``+`: تطابق الحرف السابق له 1 أو أكثر مرات.
- ``?`: تطابق الحرف السابق له 0 أو 1 مرة فقط.
- ``{n}`: تطابق الحرف السابق له n مرات.
- ``{n,}`: تطابق الحرف السابق له n أو أكثر مرة.
- ``{n,m}`: تطابق الحرف السابق له من n إلى m مرة.

4. **البداية والنهاية**:

- ``^`: تطابق البداية.
- ``\$`: تطابق النهاية.

5. **المجموعات والاستبعاد**:

- ``[abc]`: a أو b أو c تطابق أي حرف من.
- ``[^abc]`: a أو b أو c تطابق أي حرف ليس.

6. **المجموعات المتداخلة**:

أو pattern1 تطابق : `(pattern1|pattern2)` -
pattern2.

7. **الاستعلام**:

- `?:` : تطابق النمط دون حفظه كمجموعة.

هناك الكثير من المزايا والإمكانيات المتقدمة الأخرى في Regex، إلا أن هذه هي أساسيات الأوامر والعناصر الأكثر استخداماً. Regex تُستخدم بشكل واسع في البرمجة والنصوص والبحث وتحليل البيانات.