

دستورات set /a و p/ در Batch Scripting

دستور set /a

کاربرد: انجام محاسبات ریاضی و انتساب نتیجه به یک متغیر

نحوه استفاده:

Code snippet

```
set /a variable=expression
```

مثال:

Code snippet

```
set /a sum=10+20  
echo %sum%
```

```
rem Output: 30
```

نکات:

- می توانید از عملگرهای ریاضی مانند +، -، *، / و % استفاده کنید.
- می توانید از توابع ریاضی مانند sin، sqrt و cos استفاده کنید.
- می توانید از متغیرهای دیگر در عبارات خود استفاده کنید.

دستور set /p

کاربرد: خواندن ورودی از کاربر و انتساب آن به یک متغیر

نحوه استفاده:

Code snippet

```
set /p variable=prompt
```

مثال:

Code snippet

```
set /p name=What is your name?  
echo Hello, %name%!
```

```
rem Output:
```

```
rem What is your name? John Doe  
rem Hello, John Doe!
```

نکات:

- می توانید از متن دلخواه به عنوان اعلان استفاده کنید.

- می توانید از متغیرهای دیگر در اعلان خود استفاده کنید.
- می توانید از انواع داده مختلف مانند string، integer و float استفاده کنید.

ترکیب دستورات set /a و p/:

می توانید از دستورات set /a و set /p به طور همزمان برای انجام محاسبات بر اساس ورودی کاربر استفاده کنید.
مثال:

Code snippet

```
set /p radius=What is the radius of the circle?
set /a area=3.14159 * %radius%^2
echo The area of the circle is %area%.
```

rem Output:

```
rem What is the radius of the circle? 5
rem The area of the circle is 78.54.
```

در این مثال، ابتدا از کاربر خواسته می شود تا شعاع دایره را وارد کند. سپس، از شعاع برای محاسبه مساحت دایره با استفاده از فرمول πr^2 استفاده می شود. در نهایت، مساحت دایره به کاربر نمایش داده می شود.

- کد بالا بدرستی کار نخواهد کرد به دو دلیل
 - بیج اسکریپت اعداد اعشاری decimal و floating point رو پشتیبانی نمی کند.
 - عملیات توان در بیج اسکریپت در دسترس نیست.
- یادمان باشد به فاصله های عملگر ها
 - `set /a area=` در اینجا بین area و = نباید فاصله گذاشت
 - عملگر * باید با فاصله قبل و بعد لحاظ شود

با ترکیب دستورات set /a و p/ می توانید اسکریپت های Batch Scripting قدرتمند و تعاملی بسازید.

ما قصد داریم محیط دایره را با کمک batch scripting محاسبه کرده و آن را رسم کنیم.

1. قصد داریم از کاربر شعاع را دریافت کنیم.
2. فرمول $A = \pi r^2$ محیط دایره را محاسبه کنیم
3. آن را در خروجی نمایش دهیم.

برای این کار باید دستور یا باید قید دقت را برنیم یعنی 3.14 را یا 3 جایگزین کنیم و شعاع را به جای توان یکبار در خود ضرب کنیم؛

```
set /p radius=What is the radius of the circle?
set /a area=3 * %radius% * %radius%
echo The area of the circle is %area%.
```

rem Output:

```
rem What is the radius of the circle? 5
```

```
rem The area of the circle is 78.54.
```

- یا از نرم افزار calculator، VBA، پایتون و استفاده کنیم و خروجی از آنها را چاپ کنیم.

```
@echo off
```

```
rem Prompt user for radius
```

```
set /p radius=Enter the radius of the circle:
```

```
rem Call VBScript for calculation
```

```
cscript //nologo circle_area.vbs %radius% > outputBatch
```

```
SET /p MYVAR=<outputBatch
```

```
rem Call Python script with the radius as an argument
```

```
python circle_area.py %radius%
```

```
rem Display the result
```

```
echo The area of the circle is: %MYVAR%
```

```
pause
```

```
DEL outputBatch
```

جالب خواهد بود نگاهی به کد های پایتون و بچ اسکریپت هم بیاندازیم:

VBA Code for calculating Circle Area:

```
wscript.echo Round(3.14159 * WScript.Arguments(0) ^ 2, 2)
```

Python Code:

```
import sys

# Read radius from command-line argument
radius = float(sys.argv[1])

# Calculate area
area = 3.14159 * radius**2

# Print the area
print(area)
```

دستورات شرطی و حلقه‌ها

به قطعه کد زیر توجه کنید؛ این کد بازی حدس هستش؛ در آن از شرط و حلقه **until (goto)** استفاده شده است.

```
@echo off

rem Set the secret number
set /a secretNumber=10

:start
cls

echo Guess a number between 1 and 20:
set /p guess=

rem Check if the guess is correct
if %guess%==%secretNumber) %
    echo You guessed it! Congratulations!
    goto end
(

if %guess% gtr %secretNumber) %
    echo Your guess is too high. Try again.
    goto start
(

if %guess% lss %secretNumber) %
    echo Your guess is too low. Try again.
    goto start
```

```
(  
:end  
pause
```

از آنجایی که شرط پایان حلقه در دستورات **loop** از دستورات شرطی استفاده می‌کند ما نیز با آموزش دستور شرطی **if** شروع خواهیم کرد:

دستور IF در Batch Scripting

دستور IF در Batch Scripting برای بررسی شرایط و اجرای کدهای مختلف بر اساس نتیجه آن شرط استفاده می‌شود. این دستور به شما امکان می‌دهد تا برنامه‌های پیچیده‌تر و انعطاف‌پذیرتر بنویسید.

نحو کلی:

Code snippet

```
if condition (  
    code to execute if condition is true  
) else (  
    code to execute if condition is false  
)
```

مثال:

Code snippet

```
@echo off  
rem Check if a file exists  
if exist "my_file.txt" (  
    echo File exists!  
) else (  
    echo File does not exist!  
)
```

انواع شرط:

• **==**: برای مقایسه دو رشته یا عدد

- **!=**: برای بررسی عدم برابری دو رشته یا عدد
 - **<**: برای بررسی بزرگتر بودن یک عدد از عدد دیگر
 - **>**: برای بررسی کوچکتر بودن یک عدد از عدد دیگر
 - **<=**: برای بررسی بزرگتر یا مساوی بودن یک عدد از عدد دیگر
 - **>=**: برای بررسی کوچکتر یا مساوی بودن یک عدد از عدد دیگر
- عبارات مقایسه‌ای فوق درست نیست و باید از جدول زیر پیروی کرد

Operator	Description	Example
EQU	Tests the equality between two objects	2 EQU 2 will give true
NEQ	Tests the difference between two objects	3 NEQ 2 will give true
LSS	Checks to see if the left object is less than the right operand	2 LSS 3 will give true
LEQ	Checks to see if the left object is less than or equal to the right operand	2 LEQ 3 will give true
GTR	Checks to see if the left object is greater than the right operand	3 GTR 2 will give true
GEQ	Checks to see if the left object is greater than or equal to the right operand	3 GEQ 2 will give true

مثال:

Code snippet

```
@echo off
rem Check if a number is even
set /a num=10
```

```
if %num% %% 2 == 0 (
    echo Number is even
) else (
    echo Number is odd
)
```

- روبروی **if** تنها عبارات منطقی باید بیاید ؛ عبارات ریاضی با کمک **set /a** باید محاسبه شود و کد زیر به شکلی صحیح کار می‌کند:

```
@echo off
rem Check if a number is even
set /p num= enter number :
set /a booleanExp= %num% %% 2

if %booleanExp% == 0 (
    echo Number is even
) else (
    echo Number is odd
)

pause
```

- the % is a modulus operator and also a special character, hence it needs to be escaped. The escape character for % is another %.

عملگرهای منطقی:

- **&&**: برای بررسی همزمان دو شرط
- **||**: برای بررسی یکی از دو شرط

مثال:

Code snippet

```
@echo off
rem Check if a file exists and is writable
if exist "my_file.txt" && (attrib "my_file.txt" +R -A) (
    echo File exists and is writable!
```



```
) else (  
    echo File does not exist or is not writable!  
)
```

دستورات ELSE IF:

می‌توانید از دستورات ELSE IF برای بررسی شرایط بیشتر استفاده کنید.

مثال:

Code snippet

```
@echo off  
rem Check the value of a variable  
set /a grade=85  
  
if %grade% >= 90 (  
    echo Excellent!  
) else if %grade% >= 80 (  
    echo Good!  
) else if %grade% >= 70 (  
    echo Average!  
) else (  
    echo Fail!  
)
```

نکات:

- پرانتزها اختیاری هستند، اما استفاده از آنها برای خوانایی کد بهتر است.
- می‌توانید از چندین دستور IF در یک اسکریپت استفاده کنید.
- برای بررسی شرایط پیچیده‌تر، می‌توانید از دستورات GOTO و LABEL استفاده کنید.
- بایستی بجای علامت‌های مقایسه‌ای از عبارات مقایسه‌ای استفاده شود وگرنه جواب نخواهیم گرفت پس کد بالا اشتباه است و باید به کد زیر اصلاح شود.

```
@echo off  
rem Check the value of a variable
```

```
set /a grade=85

if %grade% GEQ 90 (
    echo Excellent!
) else if %grade% GEQ 80 (
    echo Good!
) else if %grade% GEQ 70 (
    echo Average!
) else (
    echo Fail!
)

pause
```

منابع:

- Batch Scripting Tutorial: [نشانی وب نامعتبر برداشته شد]
- IF Statement in Batch Scripting: [نشانی وب نامعتبر برداشته شد]

تمرین:

- یک اسکریپت بنویسید که از کاربر یک عدد می‌پرسد و سپس آن عدد را با 2 و 3 جمع می‌کند و نتیجه را نمایش می‌دهد.
- یک اسکریپت بنویسید که روز هفته را بر اساس تاریخ ورودی کاربر نمایش می‌دهد.

موفق باشید!

