

Crear un proceso en Kotlin con la clase Runtime

La clase Runtime se puede utilizar para obtener una instancia del entorno de ejecución actual a través del método estático `getRuntime()`. A partir de esta instancia, se pueden invocar diferentes métodos para realizar operaciones relacionadas con la ejecución del programa.

Para capturar y leer la salida generada por el comando ping, debemos acceder al flujo de entrada (InputStream) del proceso. Esto puede parecer confuso, pero en este caso, el flujo de entrada (InputStream) del proceso se utiliza para leer los datos que provienen de la salida generada por el comando.

```
[crayon-6a9d604111cdf257043512/]
```

Este código en Kotlin utiliza la clase Runtime para ejecutar el comando `ping google.com` y capturar la salida del proceso en tiempo real. Aquí está la explicación:

1. **Importaciones de clases necesarias:** Se importan las clases `BufferedReader` e `InputStreamReader` para leer la salida del proceso.
2. **Función principal main:** Se define la función `main`, que es el punto de entrada del programa.
3. **Obtención de una instancia de Runtime:** Se obtiene una instancia de `Runtime` utilizando el método estático `getRuntime()`.
4. **Ejecución del proceso:** Se ejecuta el comando `ping google.com` utilizando el método `exec()` de la instancia de `Runtime`. Este método devuelve un objeto `Process` que representa el proceso que se ha iniciado.
5. **Creación del lector de entrada:** Se crea un `BufferedReader` para leer la salida del proceso desde su flujo de entrada.

6. **Lectura y impresión de la salida del proceso:** Se lee cada línea de la salida del proceso utilizando el `BufferedReader` y se imprime en la consola. El bucle continúa hasta que no hay más líneas que leer.
7. **Espera a que el proceso termine:** Se espera a que el proceso termine antes de continuar. Esto asegura que el programa principal no continúe hasta que el proceso haya completado su ejecución.
8. **Impresión del código de salida del proceso:** Una vez que el proceso ha terminado, se imprime su código de salida utilizando el método `exitValue()`.