**For**

文本

AI 生成的内容可能不正确。

能使用for一行实现的代码分10行敲

就是浪费资源

（i++是变数特殊加减法（先赋值再计算），详见【PG1.2】）

（Forの実行手順：）

For( int i = 0; I < 10 ; i++){

文本

AI 生成的内容可能不正确。

注意这个i++是后置处理，第一次循环的i==0

Printf( “%d” , I ) //满足条件【I<0】运行

图片包含 杯子, 看着, 轮子, 桌子

AI 生成的内容可能不正确。

**While**

文本

AI 生成的内容可能不正确。

書き方和if文很像，满足条件才进行循环

**For**多用于决定了回数的循环

For( int i = 0; I < 10 ; i++){} ------- 管你运行什么就循环10次

**Whlie**多用于非固定回数的循环（例如ゲームループ）

While( num != 10) { scanf\_s(%d , &num) } ------- 按10结束

图示：

For：将图片包含 日程表

AI 生成的内容可能不正确。初始化、条件式（制御式）、後処理、写在一行，适合**次数明确的有限循环**；

文本

AI 生成的内容可能不正确。Whlie：结束条件隐藏在循环体内，适合**条件触发的无限循环**

**当然他们也能被用来执行一些相同的操作。（如上文例）**

|  |  |
| --- | --- |
| **while (true)** | **无限循环** |
| **while (false)** | **循环体永远不执行** |
| **while (1)** | **无限循环（因为 1 被视为 true）** |
| **while (0)** | **循环体永远不执行** |

**Do whlie**

Whlie有时因条件式连一回都不能循环

使用do while可以使得whlie至少运行一次（简单说就是倒着写，把条件式写最后先运行一帧再判断能不能循环）

用于：屏幕上有字

AI 生成的内容可能不正确。如グラフィックドライバー選択など（显卡驱动）

不管如何必须运行一次的处理

图示：

电视游戏的萤幕截图

AI 生成的内容可能不正确。

Break：用于退出循环（类似switch的break退出判断）

ゲームループ中，常与if结合使用（按键退出关卡，游戏等）