Instrukcje warunkowe, operator warunkowy, pętle

# Instrukcje warunkowe

Składnia

if ( expression ) statement-true

if ( expression ) statement-true else statement-false

Wyrażenie w if w większości wypadków może być w typie całkowitym. Wartość różna od zera to prawda, zero to fałsz. Niekiedy lepiej przyjść, że prawda to 1.

Przykłady:

#include <stdio.h>  
#include <stdlib.h>  
  
int main()  
{  
 int a =5;  
 if (a>0)  
 {  
 printf("Liczba dodania\n");  
 }  
 else  
 {  
 printf("Liczba nie jest dodania\n");  
 }  
 return 0;  
}

#include <stdio.h>  
#include <stdlib.h>  
  
int main()  
{  
 int a =-1;  
 if (a)  
 {  
 printf("Prawda\n");  
 }  
 else  
 {  
 printf("Fałsz\n");  
 }  
 return 0;  
}

#include <stdio.h>  
#include <stdlib.h>  
  
int main()  
{  
 int a =-1;  
 if (a)  
 printf("a");  
 printf("bc");  
 return 0;  
}

## Operator warunkowy

Składnia:

condition ? expression-true : expression-false

## Petla for

for ( init-clause ; cond-expression ; iteration-expression ) loop-statement

Warunek początkowy może lub nie musi mieć deklarację zmiennej interującej (od C99).

#include <stdio.h>  
#include <stdlib.h>  
  
int main()  
{  
 for(int i=1;i<5;i++)  
 {  
 printf("%d\n",i);  
 }  
 return 0;  
}

Przed C99

#include <stdio.h>  
#include <stdlib.h>  
  
int main()  
{  
 int i;  
 for(i=1;i<5;i++)  
 {  
 printf("%d\n",i);  
 }  
 return 0;  
}

Nieskończona pętla

#include <stdio.h>  
#include <stdlib.h>  
  
int main()  
{  
 for(;;)  
 {  
 printf("endless loop!");  
 }  
 return 0;  
}

Wnętrze jest opcjonalne:

#include <stdio.h>  
#include <stdlib.h>  
  
int main()  
{  
 for(int n = 0; n < 10; ++n, printf("%d\n", n))  
 ; // null statement  
 return 0;  
}

## Pętla while

while ( expression ) statement

#include <stdio.h>  
#include <stdlib.h>  
  
int main()  
{  
 int i =0;  
 while (i<5)  
 {  
 printf("%d\n",i);  
 i++;  
 }  
 return 0;  
}

#include <stdio.h>  
#include <stdlib.h>  
  
int main()  
{  
 int i =0;  
 while (i<5)  
 {  
 printf("%d\n",i);  
 if (i>2)  
 {  
 break;  
 }  
 i++;  
 }  
 return 0;  
}

#include <stdio.h>  
#include <stdlib.h>  
  
int main()  
{  
 int i =0;  
 while (i<5)  
 {  
 i++;  
 if (i==2)  
 {  
 continue;  
 }  
 printf("%d\n",i);  
 }  
 return 0;  
}

## Pętla do ... while

Składnia:

do statement while ( expression ) ;

#include <stdio.h>  
#include <stdlib.h>  
  
int main()  
{  
 int i =0;  
 do  
 {  
 printf("%d\n",i);  
 i++;  
 }  
 while (i<-2);  
 return 0;  
}

Bibliografia:

* <https://en.cppreference.com/w/c/language/if>
* <https://en.cppreference.com/w/c/language/operator_other#Conditional_operator>
* <https://en.cppreference.com/w/c/language/for>