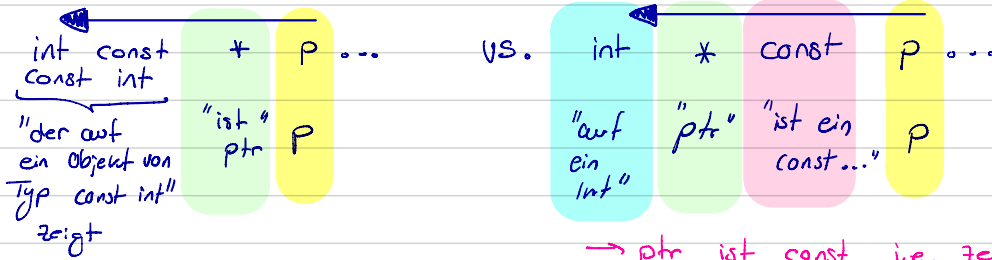


Pointer = Variable, dessen Wert eine Adresse ist.

↳ "zeigt" auf eine Stelle (i.e. Adresse) im Memory

● const (↳ von rechts nach links lesen)



→ ptr zeigt auf eine Adresse, kann aber auch auf andere Adressen zeigen

→ Inhalt der Adresse ist int und darf nicht verändert werden.

→ ptr ist const i.e. zeigt immer auf gleiche Adresse

→ Inhalt an dieser Adresse hat Typ int und darf verändert werden

● Bsp. zu pointer

```
std::cout << "pointers and references:" << std::endl;
const int &constref = 5;
std::cout << constref << std::endl;
// int &nonconstref = 5;
// error: non-const lvalue reference to type 'int' cannot bind to a temporary of type 'int'

int x = 1; // normal variable
int &y = x; // like an alias for x, same address but called y
y += 2; // were doing something with y, but y is just another name for the address of x
// so were doing something to the address of x and y and both change
std::cout << x << ", " << y << std::endl;

int *z = &x; // z is ptr to x resp the address where x lives
int *w = &y; // w is ptr to y i.e. to x

// x,y,z and *w all are the same address i.e. the same value is saved in them
std::cout << "x:" << x << ", &x:" << &x << std::endl;
std::cout << "y:" << y << ", &y:" << &y << std::endl;
std::cout << "z:" << z << ", *z:" << *z << std::endl;
std::cout << "w:" << w << ", *w:" << *w << std::endl;
```

} Nicht verwechseln: pointer und referenzen sind nicht das gleiche!
 e.g. für mehr:

<https://stackoverflow.com/questions/57483/what-are-the-differences-between-a-pointer-variable-and-a-reference-variable>

output:

```
pointers and references:
5
3, 3
x:3, &x:0x7ff7bc0fc468
y:3, &y:0x7ff7bc0fc468
z:0x7ff7bc0fc468, *z:3
w:0x7ff7bc0fc468, *w:3
```

|| → alle haben die gleiche Adresse

memory Adressen sind einfach hex zahlen

● Pointer dereferenzieren ≙ dorthin gehen, wo der ptr hinzeigt.

↳ nicht mit referenzen verbinden

```
e.g. int x = 1;
int *z = &x;
```

↑ Adress operator: Wert eines ptr muss ja eine Adresse sein.

// Wert an stelle im memory, an die z zeigt ausgeben:

```
std::cout << *z ..... ;
```

(Für die, die es interessiert: hier mehr info allg. zu ptr / refs)