Recall: ++ for Variables

Recall: ++ for Variables

```
++int_var
```

```
int& pre_incr (int& a)
{
    a = a + 1;
    return a;
}
```

Recall: ++ for Variables

```
++int_var
```

```
int& pre_incr (int& a)
{
    a = a + 1;
    return a;
}
```

Increment value of `a`.
Then return **new** value as **l-value**.

Recall: ++ for Variables

```
++int_var
```

```
int_var++
```

```
int& pre_incr (int& a)
{
    a = a + 1;
    return a;
}
```

```
int post_incr (int& a)
{
    int value = a;
    a = a + 1;
    return value;
}
```

Increment value of a.
Then return **new** value as **l-value**.

Recall: ++ for Variables

```
++int_var
```

```
int& pre_incr (int& a)
{
    a = a + 1;
    return a;
}
```

Increment value of a.
Then return **new** value as **l-value**.

```
int_var++
```

```
int post_incr (int& a)
{
    int value = a;
    a = a + 1;
    return value;
}
```

Increment value of a.
Then return **old** value as **r-value**.

Recall: ++ for Variables

++int_var

```
int& pre_incr (int& a)
{
    a = a + 1;
    return a;
}
```

Increment value of a.
Then return **new** value as **l-value**.

int_var++

```
int post_incr (int& a)
{
    int value = a;
    a = a + 1;
    return value;
}
```

Increment value of a.
Then return **old** value as **r-value**.

++ for Pointers

++ for Pointers

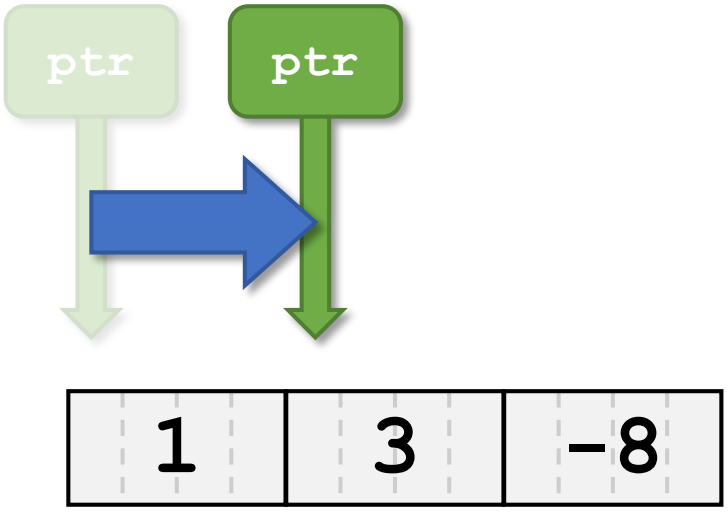
- Same idea...

++ for Pointers

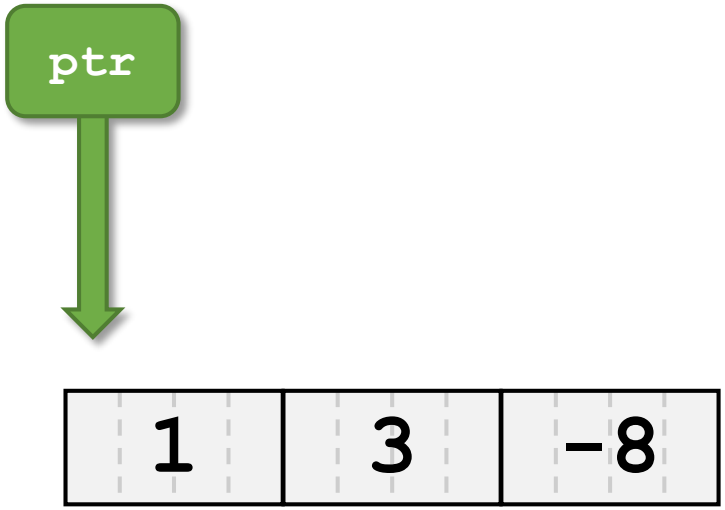
- Same idea...
- …but: value of pointer is an **address**.

++ for Pointers

- Same idea...
- …but: value of pointer is an **address**.
  → Shift pointer to **next object**.

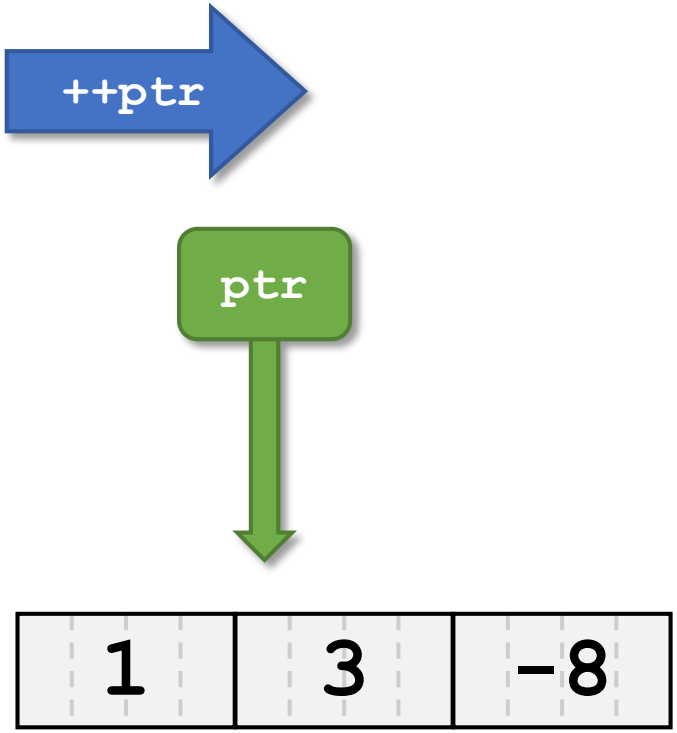


++ptr and ptr++



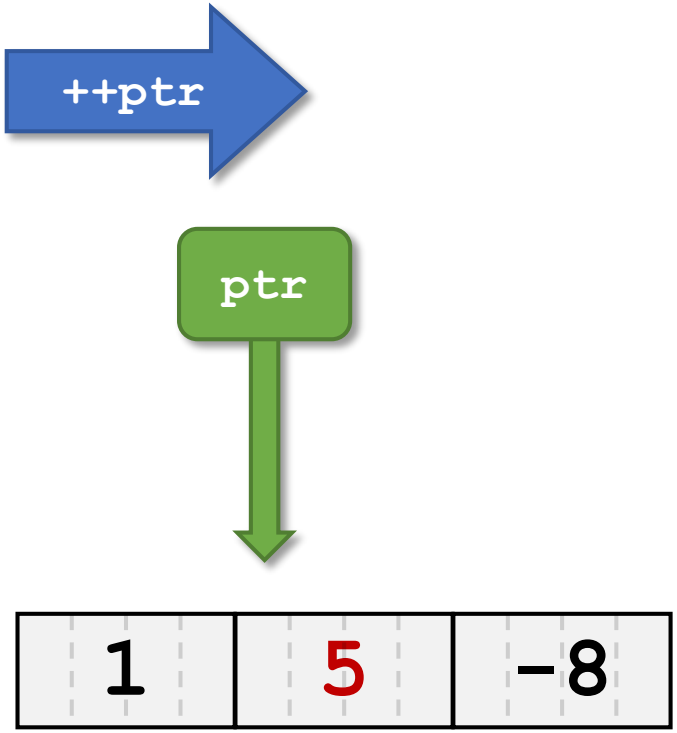

```
*(++ptr) = 5;
```

++ptr and ptr++




```
*(++ptr) = 5;
```

++ptr and ptr++

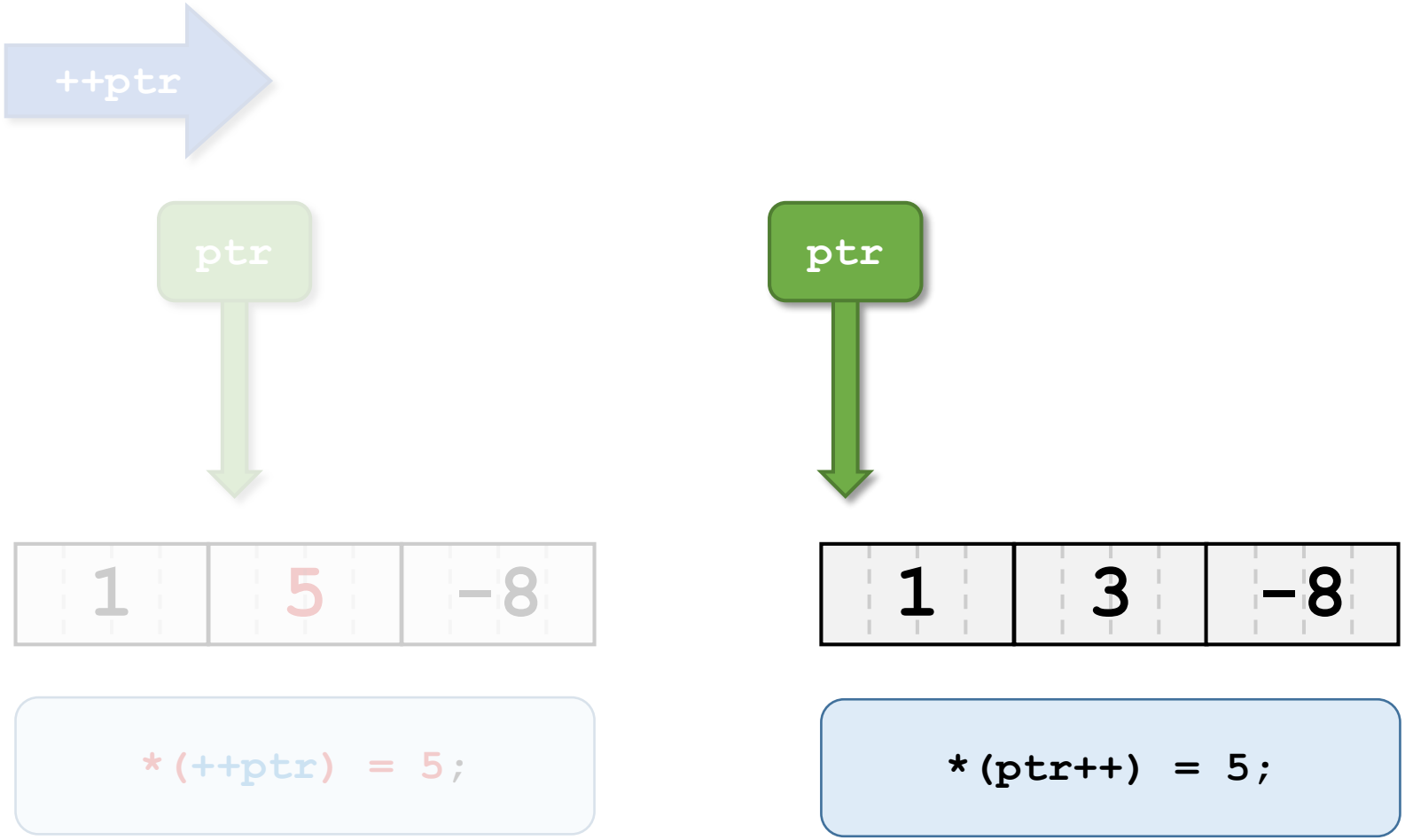



```
*(++ptr) = 5;
```

++ptr and ptr++

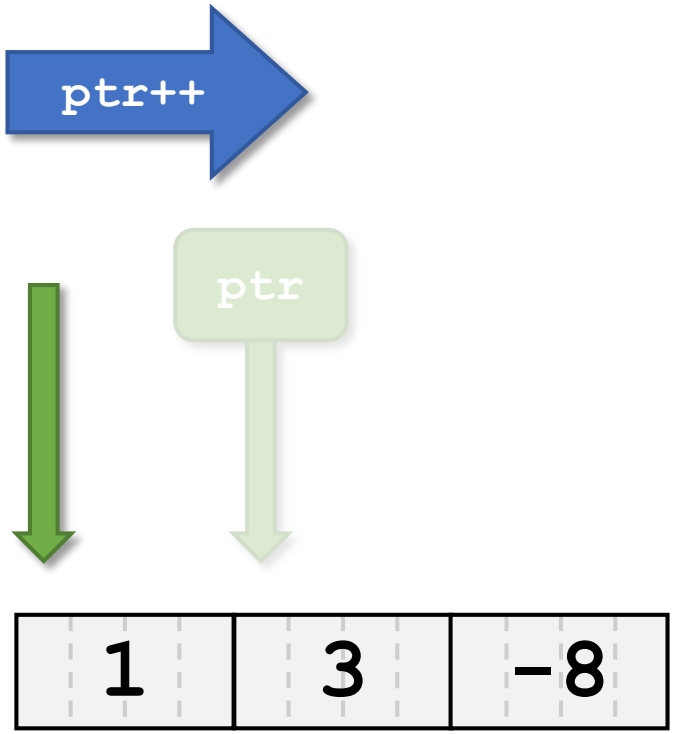

```
*(++ptr) = 5;
```

++ptr and ptr++



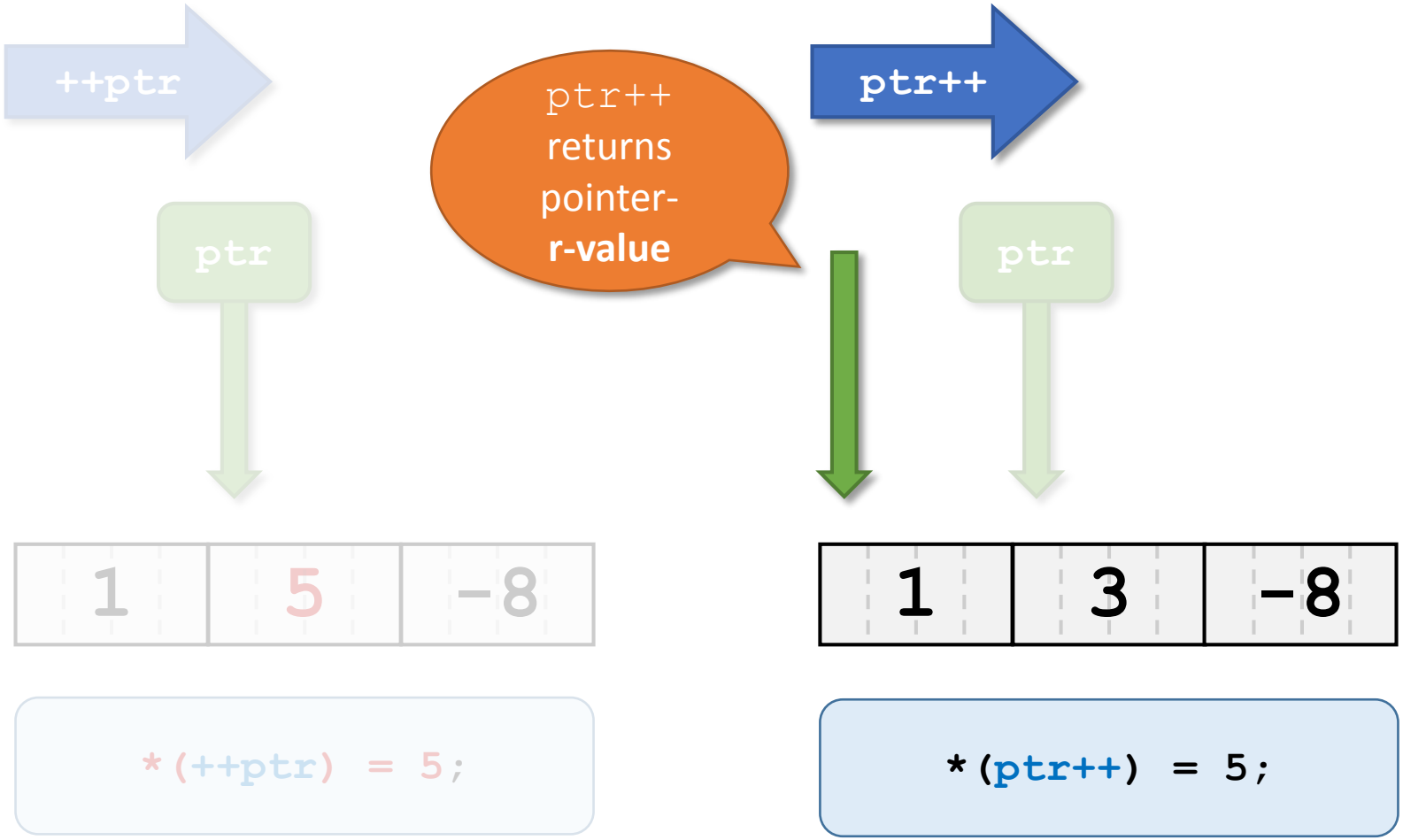
++ptr and ptr++


```
*(++ptr) = 5;
```

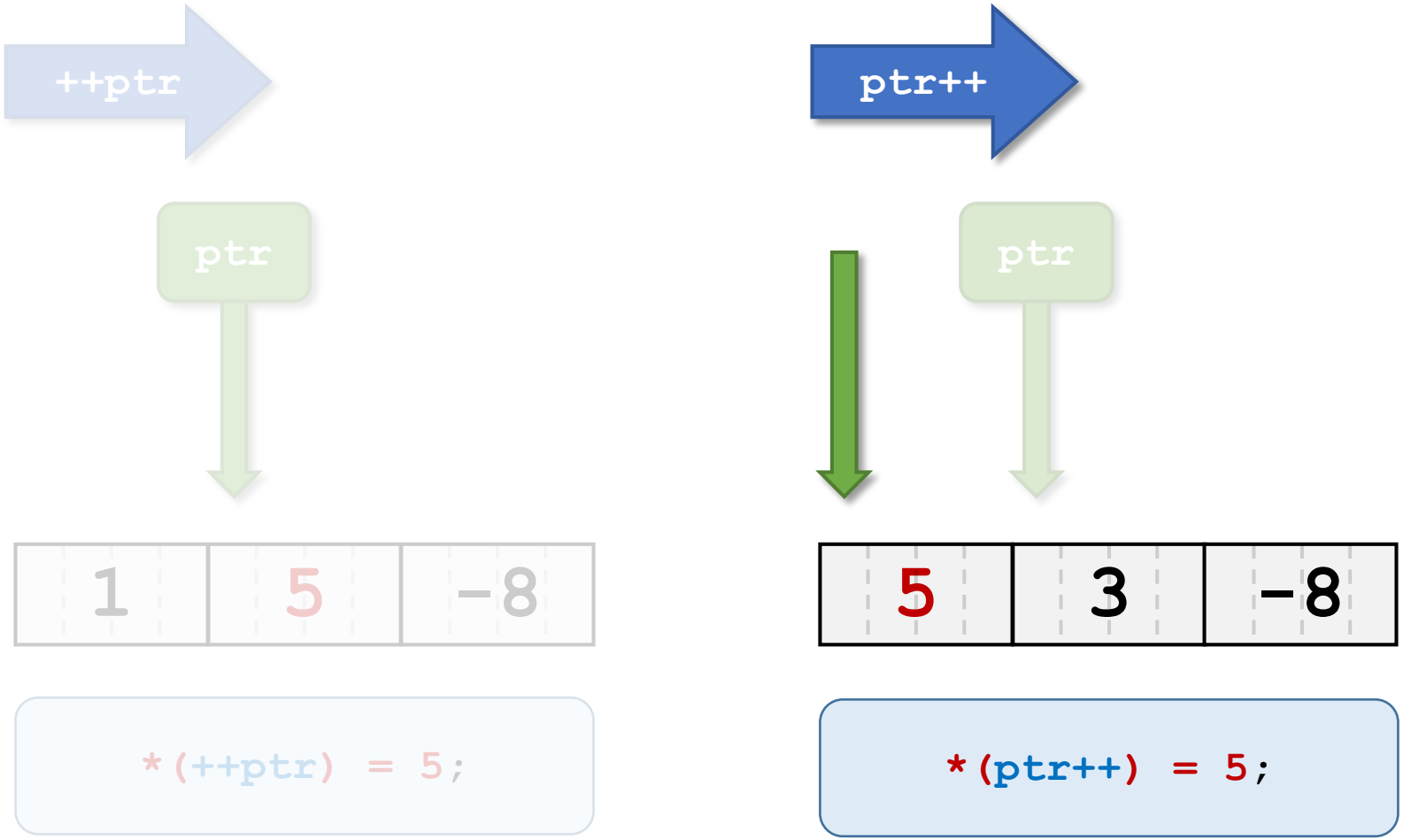
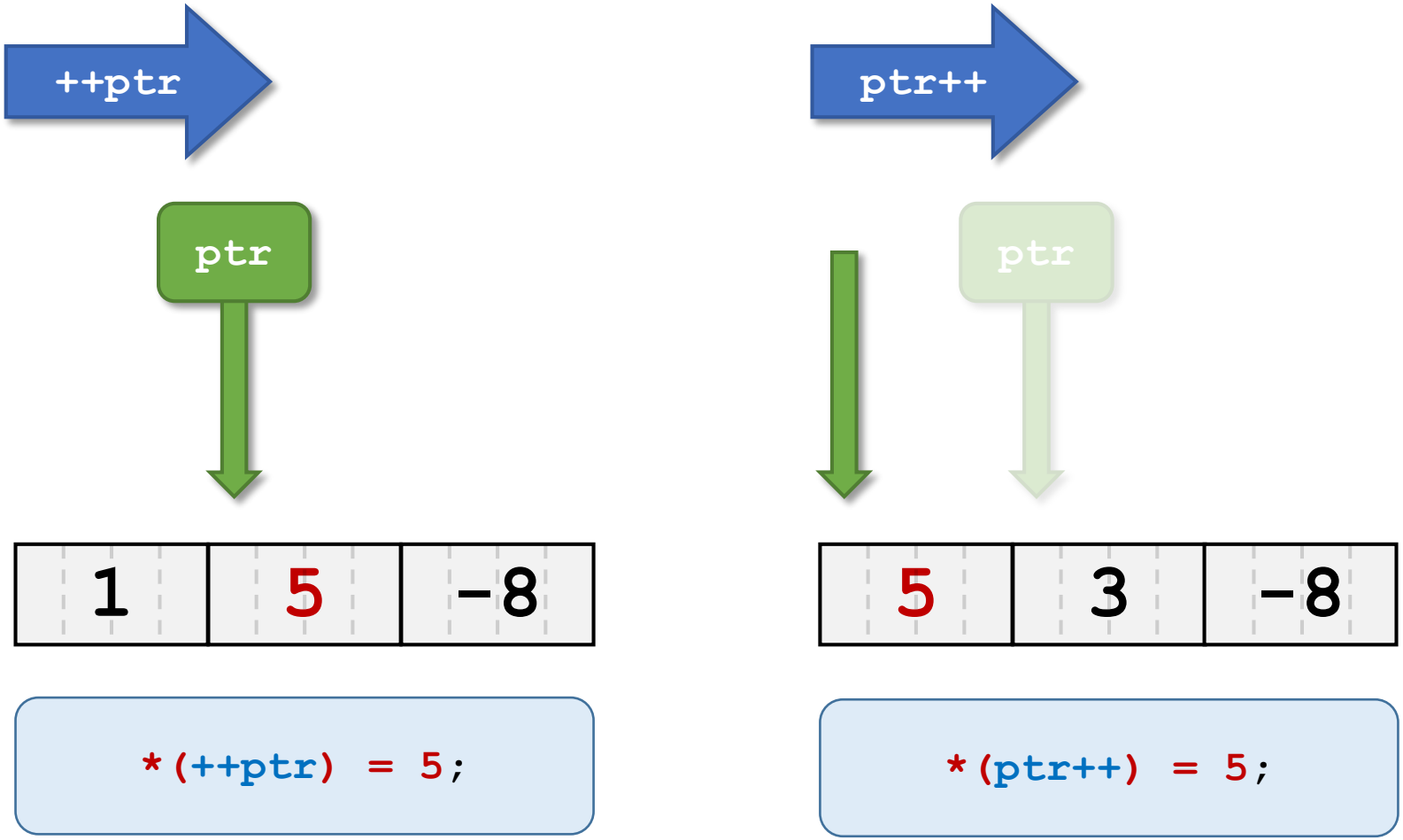



```
*(ptr++) = 5;
```

++ptr and ptr++



++ptr and ptr++



++ptr and ptr++