

### cast-basic

```
out = (*int32)(unsafe.Pointer(in))
```

### cast-bytes

```
return ((*[10]byte)(unsafe.Pointer(x)))[:]
```

### cast-struct

```
out = (*runtime.Unknown)(unsafe.Pointer(in))
```

### cast-header

```
hdr := &reflect.SliceHeader{
    Data: uintptr(unsafe.Pointer(&data[i])),
    Len:  42,
    Cap:  42,
}
retVal = append(retVal, *(*[]uint8)(unsafe.Pointer(hdr)))
```

### cast-pointer

```
return unsafe.Pointer(ptr)
```

### pointer-arithmetic

```
unaligned := uintptr(unsafe.Pointer(&value[0])) & 3
```

### delegate

```
func (encoder *Encoder) Encode(ptr unsafe.Pointer) {
    encoder.UnsafeIndirect(ptr)
}
```

### memory-access

```
deReferenced := *((*unsafe.Pointer)(ptr))
```

### syscall

```
n, _, errno := syscall.Syscall(syscall.SYS_RECVMSG, s,
    uintptr(unsafe.Pointer(h)), uintptr(flags))
```

### definition

```
type unsafeType struct {
    ptr unsafe.Pointer
}
```

### unused

```
func Encode(ptr unsafe.Pointer, stream *Stream) {
    stream.WriteEmptyArray()
}
```