Rotman

EXTRACTING ELEMENTS

February 8, 2022 Prepared by Niti
TDMDAL & FinHUB

Rotman School of Management
UNIVERSITY OF TORONTO

Extracting Elements

1. Indexing
2. Slicing
3. Stepping
4. Negative Indexing

Extracting Elements

Indexing	• **Extract only one element at a time** • **[position]**
Slicing	• **Extract more than one element at time** • **[start:stop]**
Stepping	• **Extract every n^{th} element** • **[start:stop:step]**
Negative Indexing	• **Extract elements from the end of the list counting backwards**

- Indexing, slicing and stepping is a way of manipulating selective elements from a given sequence
- Most python tools follow the same convention of indexing, slicing and stepping
- Let's extract some elements from the variable b below -

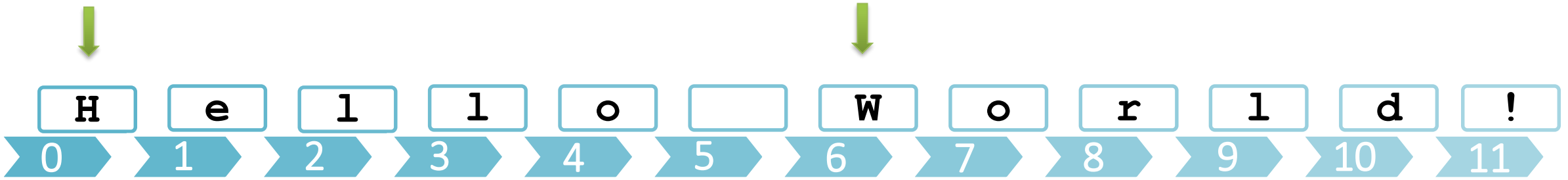
```
b = 'Hello World!'
```

Rotman

Extracting Elements : Indexing

```
print(b[0])
→ 'H'
```

- ✓ Python indexing starts at 0
- ✓ Index operator [ ] is used for indexing
- ✓ Think of index as a pointer between the elements




```
print(b[6])
→ 'W'
```


Rotman

Extracting Elements : Slicing

```
print(b[6:10])
'Worl'
```

What is after the start index?

What is before the stop index?

- ✓ Index operator with a colon (:) is used for slicing
- ✓ Start and stop index can be defined on either side of the colon
- ✓ Stop index defines where to stop extracting elements

Rotman

Extracting Elements : Slicing

```
print(b[6: ])
→'World!'
```

Everything after the start index

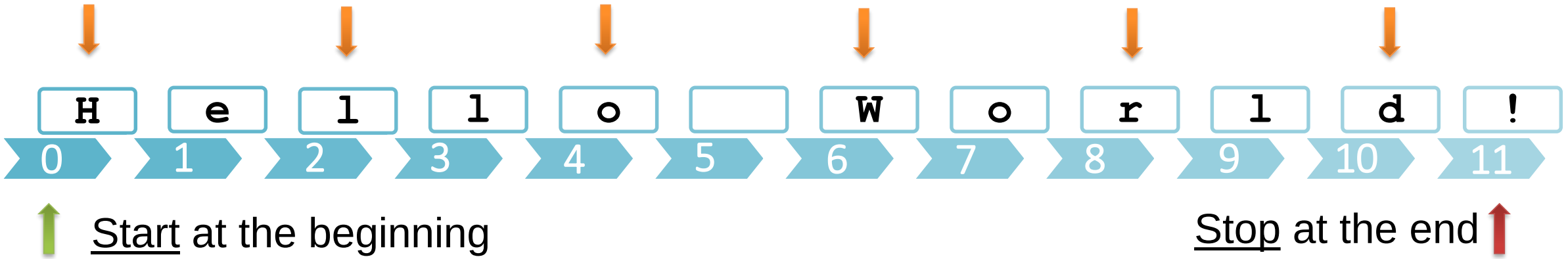
Rotman

Extracting Elements : Stepping

```
print(b[::2 ])
→'HloWrd'
```

✓ Index operator with double colon `[::]` is used for stepping

Select every other element
between start and stop indices



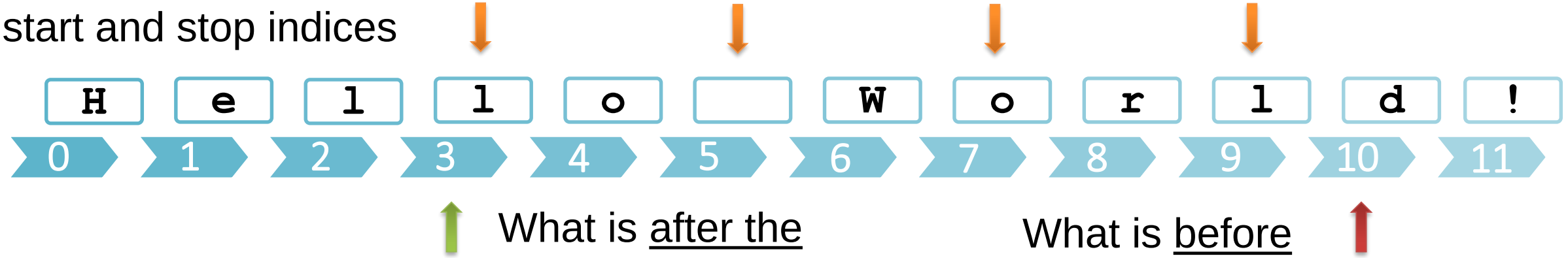
Rotman

Extracting Elements : Slicing with Stepping

```
print(b[3:10:2])
→'l ol'
```

✓ Start, stop and step index can be defined on each sides of the colons

Every other element between start and stop indices



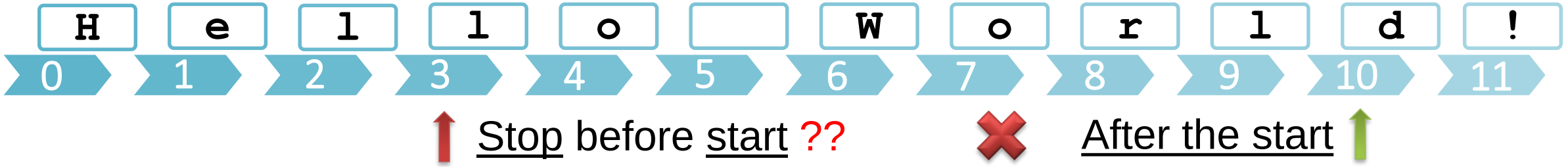
Rotman

Extracting Elements : Slicing with Stepping

```
print(b[10:3:2])
→ ''
```

✓ We can use negative index to go backwards

Does NOT work because
elements are extracted
from right of the start position
up to the left of stop position

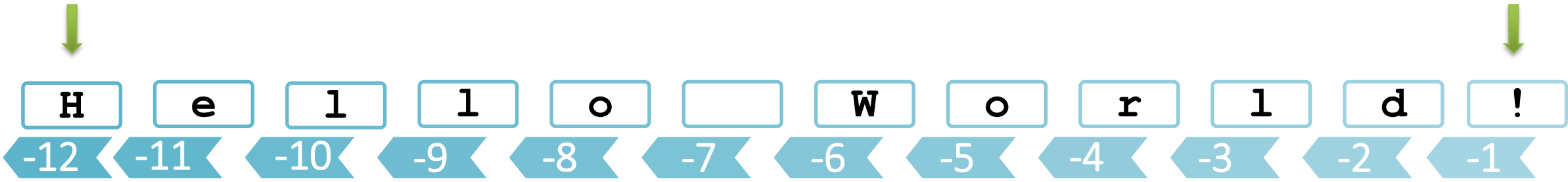


Rotman

Extracting Elements : Negative Indexing

```
print(b[-12])
```
→ `'H'`

✓ Negative index does not have 0



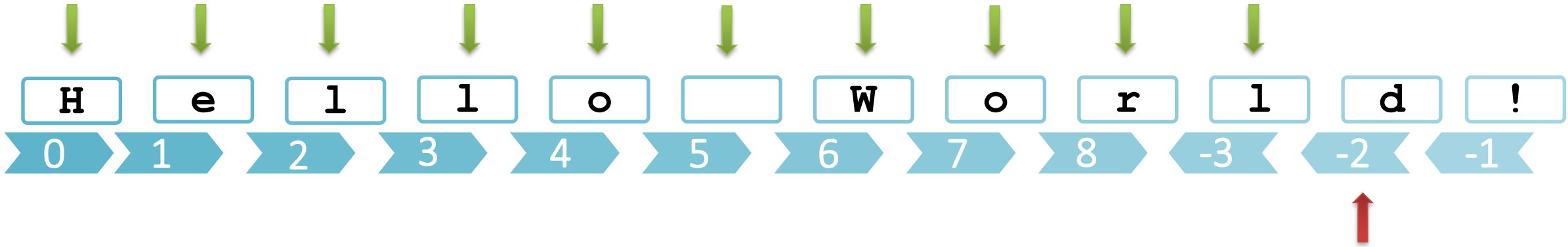

```
print(b[-1])
```
→ `'!'`

Rotman

Extracting Elements : Slicing with Negative Index

```
print(b[:-2])
→'Hello Worl'
```

Everything except last two items



Rotman

Extracting Elements : Slicing with Negative Index

```
print(b[-2:])
→'d!'
```

Everything after start

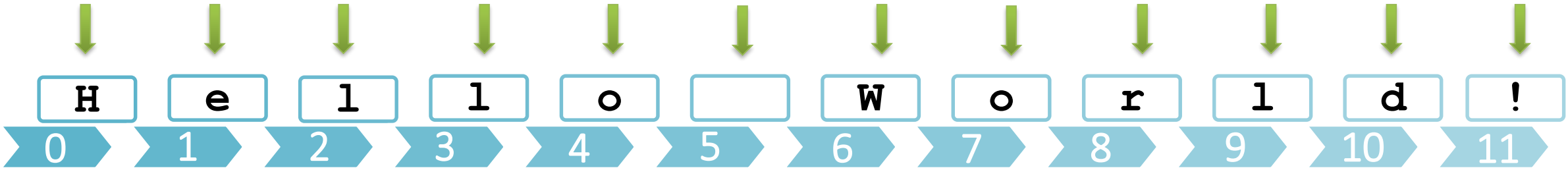
What is after the start index?

Rotman

Extracting Elements : Slicing and Stepping with Negative Index

```
print(b[::-1])
→'!dlroW olleH'
```

Everything, reversed

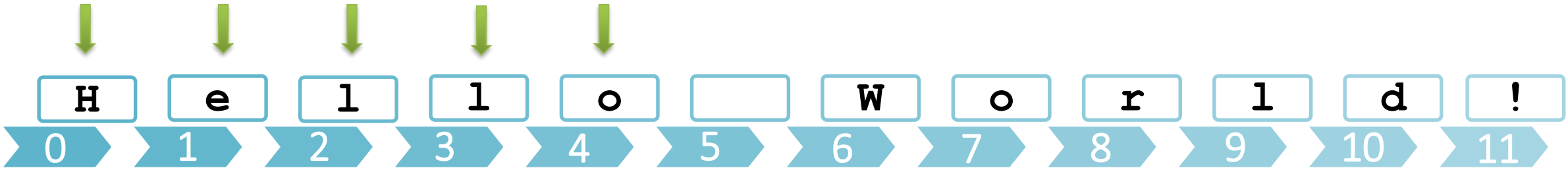


Rotman

Extracting Elements : Slicing and Stepping with Negative Index

```
print(b[4::-1])
→'olleH'
```

First five elements, reversed



Rotman

Extracting Elements : Slicing and Stepping with Negative Index

```
print(b[:-3:-1])
→'!d'
```

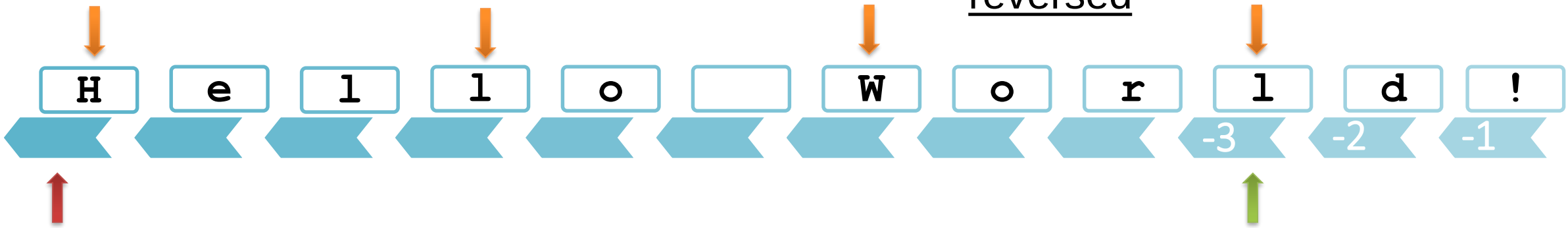
Last two items, reversed

Rotman

Extracting Elements : Slicing and Stepping with Negative Index

```
print(b[-3::-3])
→'lWlH'
```

Every two other element,
except the last two elements,
reversed



Rotman