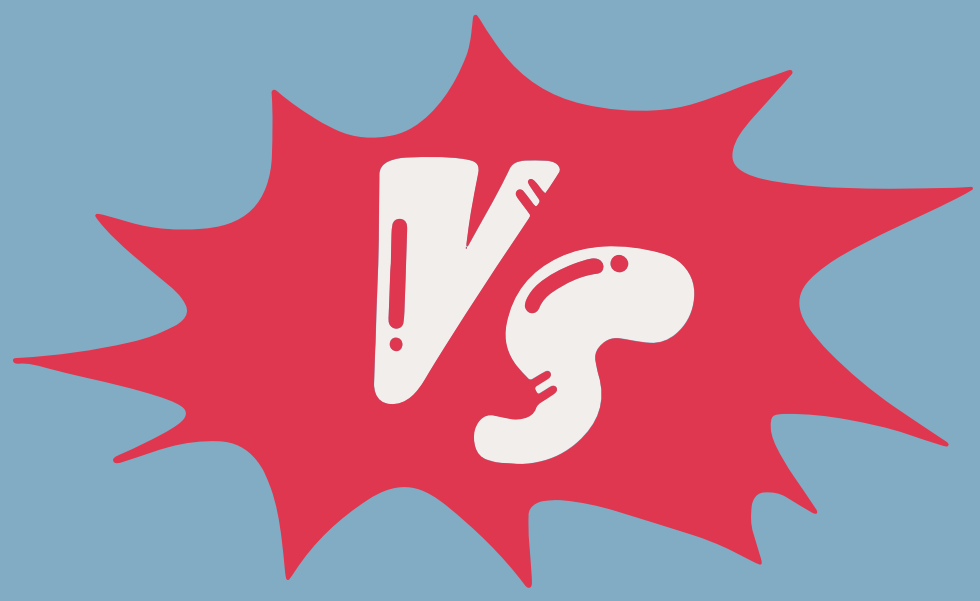

```
arrayList.OfType<string>();
```




```
arrayList.Cast<string>();
```

@romain-od 

www.code-review.tech

Cast<T>()

Implementation

```
public IEnumerable<T> Cast<T>(this IEnumerable source)
{
  foreach(object o in source)
    yield return (T) o;
}
```

You should call Cast<string>() if you know that all of the items are strings.
If some of them aren't strings, you'll get an exception.

@romain-od
www.code-review.tech

OfType<T>()

Implementation

```
public IEnumerable<T> OfType<T>(this IEnumerable source)
{
  foreach(object o in source)
    if(o is T t)
      yield return t;
}
```

You should call OfType<string>() if you know that some of the items aren't strings and you don't want those items.
If some of them aren't strings, they won't be in the new IEnumerable<string>.

@romain-od

www.code-review.tech

TL;DR

OfType - return only the elements that can safely be cast to type x.

Cast - will try to cast all the elements into type x. if some of them are not from this type you will get InvalidCastException

```
object[] objs = new object[] { "12345", 12 };
objs.Cast<string>().ToArray(); //throws InvalidCastException
objs.OfType<string>().ToArray(); //return { "12345" }
```

@romain-od

www.code-review.tech

THANKS FOR READY

SHARE YOUR THOUGHTS

FOLLOW ME FOR MORE SHARING

@romain-od

www.code-review.tech