# Wrapper class Integer

<table>
<thead>
<tr>
<th>Integer</th>
<th>234</th>
</tr>
</thead>
<tbody>
<tr>
<td>intValue()</td>
<td></td>
</tr>
<tr>
<td>byteValue()</td>
<td></td>
</tr>
<tr>
<td>shortValue()</td>
<td></td>
</tr>
<tr>
<td>longValue()</td>
<td></td>
</tr>
<tr>
<td>floatValue()</td>
<td></td>
</tr>
<tr>
<td>doubleValue()</td>
<td></td>
</tr>
<tr>
<td>toString()</td>
<td></td>
</tr>
<tr>
<td>equals(Object)</td>
<td></td>
</tr>
</tbody>
</table>

The `int` field can’t be changed — it is **immutable**.

**Reason for wrapper class Integer**: to be able to handle an `int` value as an object.

**Reason for wrapper class Integer**: to have a place to put constants and static methods that deal with int values.
Wrapper class Integer

The int field can’t be changed — it is **immutable**.

**Reason for wrapper class Integer**: to be able to handle an int value as an object.

**Reason for wrapper class Integer**: to have a place to put constants and static methods that deal with int values.
## Wrapper class Integer

### static components
- `MIN_VALUE`
- `MAX_VALUE`

### Methods
- `intValue()`
- `byteValue()`
- `shortValue()`
- `longValue()`
- `floatValue()`
- `doubleValue()`
- `toString()`
- `equals(Object)`
- `toBinaryString(int)`
- `toOctalString(int)`
- `toHexString(int)`
- `parseInt(String)`

The **int** field can’t be changed — it is **immutable**.

**Reason for wrapper class Integer**: to be able to handle an **int** value as an object.

**Reason for wrapper class Integer**: to have a place to put constants and static methods that deal with int values.