

Joining Lines and Literals

IntelliJ IDEA makes it possible to concatenate two unselected or several selected lines into one, removing the extra spaces, and providing the proper syntax. This operation smartly analyzes the lines being joined and treats them accordingly. For example, you can join lines of code, lines of comments, field declaration and initialization.

In this section:

- [Joining lines](#)
- [Joining string literals](#)
- [Examples](#)

To smartly join several lines

1. Place the caret on the line, to which the other lines should be added.

```
141 foo(a,  
142     b,  
143     c,  
144     str);
```

2. Sequentially press `Ctrl+Shift+J` keyboard shortcut, until all fragments are joined in a single line.

```
141 foo(a, b, c,  
142     str);  
  
141 foo(a, b, c, str);
```

You can select the lines and press `Ctrl+Shift+J` to obtain the same result.

To smartly join several literals into a single string

1. Select the lines with string literals that should be joined.

```
153 String theBest = new String("IntelliJ IDEA is " +  
154     "the most intelligent " +  
155     "IDE around");
```

2. Press `Ctrl+Shift+J` keyboard shortcut. All redundant characters (spaces, quotes, and plus signs) are gone.

```
153 String theBest = new String("IntelliJ IDEA is the most intelligent IDE around");
```

Examples

Joining a field or variable declaration and assignment:

```
140 int myInt;  
141 myInt = 1;
```

Pressing `Ctrl+Shift+J` produces the following result, with the unwanted spaces and variable name in the second line removed:

```
140 int myInt = 1;
```

Consider the following pair of statements:

```
46 int intVar_1 = START_RATE;  
47 intVar_1 -= END_RATE;
```

Press Ctrl+Shift+J to join these lines into a correct single-line statement.

```
46 int intVar_1 = START_RATE - END_RATE;
```

See Also

Getting Started:

- [Basic Editing Procedures](#)

Web Resources:

- [Developer Community](#) 