扩展已有字段

为了扩展现有字段，您必须将内置字段（例如FieldTextInput，FieldColour）子类化，然后修改其中的一部分以满足您的需求。 您可以修改的字段的某些部分是：

- 它的编辑器
- 它的块显示
- 它显示的文本

如果要创建不需要任何内置字段行为的自定义字段，则应将Field子类化。

常用扩展

大多数自定义字段扩展了以下三种类型之一：

1. 文本输入

   如果希望用户在字段中键入内容，则应扩展FieldTextInput。

2. 数字

   如果要存储数字，则应扩展FieldNumber。

3. 下拉菜单

   如果要创建一个下拉菜单，但希望它存储与默认字符串模型不同的模型，则应扩展FieldDropdown。

   错误警告：在扩展FieldDropdown之前，请检查下拉字段的自定义选项是否不能满足您的需求。

在某些情况下，您可能希望扩展其他字段类型。 例如，FieldLabelSerializable扩展了FieldLabel。

子类化

```
'use strict';

goog.provide('MySubclassName');

goog.require('MySuperclassName');

MySubclassName = function(opt_value, opt_validator) {
  opt_value = this.doClassValidation_(opt_value);
  if (opt_value === null) {
    opt_value = MySubclassName.DEFAULT_VALUE;
  }  // Else the original value is fine.

 MySubclassName.superClass_.constructor.call(this, opt_value, opt_validator);
};
goog.inherits(MySubclassName, MySuperclassName);
```

字段的子类的构造函数看起来与自定义字段的构造函数非常相似。 只需将MySubclassName替换为新字段的名称，并将MySuperclassName替换为要子类化的字段的名称（例如Blockly.FieldWhatever）。 子构造函数的签名通常应与超级构造函数的签名匹配。

JSON和注册

您还应该确保注册该字段，以便它与JSON格式一起使用。

```
MySubclassName.fromJson = function(options) {
  var value = Blockly.utils.replaceMessageReferences(options['value']);
  return new MySubclassName(value);
};

Blockly.Field.register('field_my_subclass', MySubclassName);
```

注册扩展字段与注册自定义字段相同。 只需将MySubclassName替换为新字段，然后将'field_my_subclass'替换为所需的JSON字段名称。