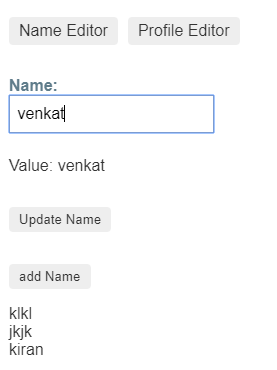
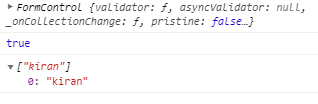
**Reactive Forms**

Only one text box can handle with only FormControl

<label>

  Name:

  <input type="text" [formControl]="name">

</label>

<p>

  Value: {{ name.value }}

</p>

<p>

  <button (click)="updateName()">Update Name</button>

</p>

<p>

  <button (click)="addNames()">add Name</button>

</p>

<div \*ngFor="let names of namesArray">{{names}}</div>

import { Component } from '@angular/core';

import { FormControl, Validators } from '@angular/forms';

@Component({

  selector: 'app-name-editor',

  templateUrl: './name-editor.component.html',

  styleUrls: ['./name-editor.component.css']

})

export class NameEditorComponent {

  name = new FormControl('',[Validators.required]);

  namesArray=[];

  updateName() {

    this.name.setValue('Nancy');

  }

  addNames(){

    console.log(this.name)

    console.log(this.name.valid)

    this.namesArray.push(this.name.value)

    console.log(this.namesArray);

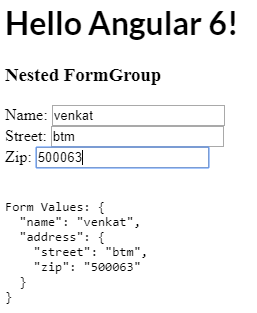
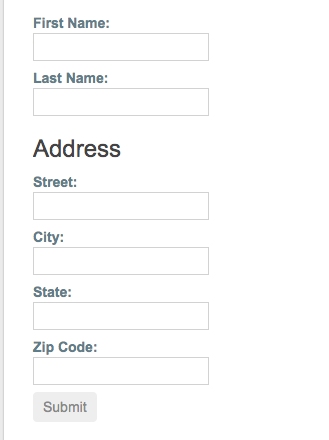
    this.name.reset()

  }

}

## Creating nested form groups

Address was nested



import { [Component](https://angular.io/api/core/Component) } from '@angular/core';

import { [FormGroup](https://angular.io/api/forms/FormGroup), [FormControl](https://angular.io/api/forms/FormControl) } from '@angular/forms';

@[Component](https://angular.io/api/core/Component)({ selector: 'app-profile-editor',

[templateUrl](https://angular.io/api/core/Component#templateUrl): './profile-editor.component.html',

[styleUrls](https://angular.io/api/core/Component#styleUrls): ['./profile-editor.component.css']

})

export class ProfileEditorComponent {

profileForm = new [FormGroup](https://angular.io/api/forms/FormGroup)({

firstName: new [FormControl](https://angular.io/api/forms/FormControl)(''),

lastName: new [FormControl](https://angular.io/api/forms/FormControl)(''),

address: new [FormGroup](https://angular.io/api/forms/FormGroup)({ street: new [FormControl](https://angular.io/api/forms/FormControl)(''),

city: new [FormControl](https://angular.io/api/forms/FormControl)(''),

[state](https://angular.io/api/animations/state): new [FormControl](https://angular.io/api/forms/FormControl)(''),

zip: new [FormControl](https://angular.io/api/forms/FormControl)('')

}) }); }

Html

<div formGroupName="address">

<h3>Address</h3>

<label>

Street:<input type="text" formControlName="street">

</label>

<label>

City: <input type="text" formControlName="city">

</label>

<label>

State:<input type="text" formControlName="state">

</label>

<label>

Zip Code:<input type="text" formControlName="zip">

</label>

</div>

## Generating form controls with FormBuilder

<https://angular.io/guide/reactive-forms#creating-nested-form-groups>

import { [Component](https://angular.io/api/core/Component) } from '@angular/core';

import { [FormBuilder](https://angular.io/api/forms/FormBuilder) } from '@angular/forms';

@[Component](https://angular.io/api/core/Component)({ selector: 'app-profile-editor',

[templateUrl](https://angular.io/api/core/Component#templateUrl): './profile-editor.component.html',

[styleUrls](https://angular.io/api/core/Component#styleUrls): ['./profile-editor.component.css']

})

export class ProfileEditorComponent {

profileForm = this.fb.group({

firstName: [''], lastName: [''],

address: this.fb.group({

street: [''],

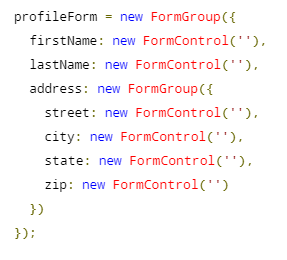
city: [''],

[state](https://angular.io/api/animations/state): [''],

zip: [''] }), });

constructor(private fb: [FormBuilder](https://angular.io/api/forms/FormBuilder)) { } }

creating form Builder



<https://stackblitz.com/angular/jlqxjyojodn?file=src%2Fapp%2Fprofile-editor%2Fprofile-editor.component.ts>

patchValues

 updateProfile() {

    this.profileForm.patchValue({

      firstName: 'Nancy',

      address: {

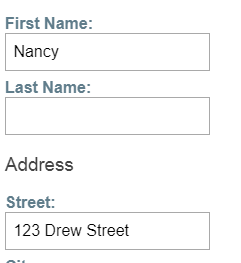
        street: '123 Drew Street'

      }

    });

  }

Auto fill update with patch values

Form Group , nested form, form Array

## Dynamic controls using form arrays

import { Component } from '@angular/core';

import { FormBuilder } from '@angular/forms';

import { Validators } from '@angular/forms';

import { FormArray } from '@angular/forms';

@Component({

  selector: 'app-profile-editor',

  templateUrl: './profile-editor.component.html',

  styleUrls: ['./profile-editor.component.css']

})

export class ProfileEditorComponent {

  profileForm = this.fb.group({

    firstName: ['', Validators.required],

    lastName: [''],

    address: this.fb.group({

      street: [''],

      city: [''],

      state: [''],

      zip: ['']

    }),

    aliases: this.fb.array([

      this.fb.control('')

    ])

  });

  get aliases() {

    return this.profileForm.get('aliases') as FormArray;

  }

  constructor(private fb: FormBuilder) { }

updateProfile() {

    this.profileForm.patchValue({

      firstName: 'Nancy',

      address: {

        street: '123 Drew Street'

      }

    });

  }

  addAlias() {

    this.aliases.push(this.fb.control(''));

  }

  onSubmit() {

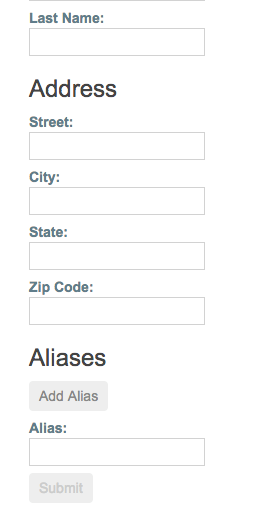
    // TODO: Use EventEmitter with form value

    console.warn(this.profileForm.value);

  }

}

<https://angular.io/guide/reactive-forms#reactive-forms-api>



<div [**formArrayName**](https://angular.io/api/forms/FormArrayName)**="aliases">**

<h3>Aliases</h3>

<button (click)="addAlias()">Add Alias</button>

<div \*[ngFor](https://angular.io/api/common/NgForOf)="let alias of aliases.controls; let i=index">

<!-- The repeated alias template -->

<label> Alias: <input type="text" [[formControlName](https://angular.io/api/forms/FormControlName)]="i"> </label>

</div>

</div>