

Les éléments à télécharger sont disponibles à l'adresse suivante :
<http://www.editions-eni.fr>
Saisissez la référence ENI de l'ouvrage **RITYP** dans la zone de recherche et validez. Cliquez sur le titre du livre puis sur le bouton de téléchargement.

Avant-propos

Chapitre 1 Introduction

- 1. Un peu d'histoire. 9
- 2. ECMAScript 13
- 3. Pourquoi TypeScript? 17
 - 3.1 Transpilation 18
 - 3.2 Typage statique 21
- 4. Les dessous du typage. 23
 - 4.1 Les fichiers de définition 23
 - 4.2 Typage structurel 24
- 5. L'architecture de TypeScript 27
 - 5.1 Core TypeScript Compiler. 28
 - 5.2 Standalone TS Compiler 28
 - 5.3 Language Service. 28
 - 5.4 Options de compilation 29
- 6. Environnement de développement 30
 - 6.1 Node.js. 30
 - 6.2 Visual Studio Code 32
 - 6.3 Installer TypeScript 38

Chapitre 2

Types et instructions basiques

1. Variable et portée	41
2. Types basiques	45
2.1 Introduction	45
2.2 Types primitifs basiques	45
2.3 Types primitifs : undefined et null	49
2.4 Types par référence : array et tuples	52
2.5 Types par référence : object	54
2.6 Types spéciaux : any et void	56
2.7 Affirmation de type	58
2.8 Type spécial : unknown	59
3. Décomposition	62
4. Énumération	65
5. Conditionnelle	72
6. Opérateurs	74
6.1 Opérateurs arithmétiques	74
6.2 Opérateurs unaires	75
6.3 Opérateurs de comparaison	76
6.4 Opérateurs logiques combinatoires	76
7. Zoom sur les opérateurs	77
7.1 Opérateurs unaires + et -	77
7.2 Opérateur unaire !	78
7.3 Opérateurs == et ===	78
7.4 Opérateurs et &&	79
7.5 Opérateur de décomposition	81
8. Boucles	82
8.1 Boucles for...in	83
8.2 Boucle for...of	84
9. Symbol	85

- 10. Itération et collections 90
 - 10.1 Iterator 90
 - 10.2 Générateurs 92
 - 10.3 Map et Set 92
 - 10.3.1Map 92
 - 10.3.2Set 93
 - 10.3.3WeakMap et WeakSet 94
- 11. Fonctions 95
 - 11.1 Les bases 95
 - 11.2 Paramètre rest 97
 - 11.3 Les closures 98
 - 11.4 This et les fonctions fléchées 100
 - 11.5 Type fonction 104
- 12. Prototype 105
- 13. Gestion des exceptions 106

Chapitre 3
La programmation orientée objet

- 1. Introduction 109
- 2. Les classes 110
- 3. Les propriétés 113
- 4. Les méthodes 116
- 5. Les constructeurs 122
- 6. Le statisme 124
- 7. L'accessibilité des membres 126
- 8. L'encapsulation 132
- 9. L'héritage 135
- 10. L'abstraction 142
- 11. Les interfaces 145

12. Le polymorphisme	156
13. Les principes SOLID	158
13.1 Introduction aux principes SOLID	158
13.2 Le principe de responsabilité unique	158
13.3 Le principe d'ouverture à l'extension, fermeture à la modification	161
13.4 Le principe de substitution de Liskov	164
13.5 Le principe de ségrégation des interfaces	169
13.6 Le principe d'inversion des dépendances	171

Chapitre 4

Les modules

1. Introduction	175
2. Historique	176
2.1 Pattern module	176
2.2 AMD et CommonJs	178
2.3 Standardisation	183
3. La norme ECMAScript 2015	183
3.1 Export	187
3.2 Import	191
4. Chargement et résolutions	194
4.1 Résolutions	194
4.2 Chargement	197

Chapitre 5

La généricité

1. Introduction	199
2. Déclaration de base	199
3. Classes et interfaces	202
4. Contraintes	205

5. Générique et constructeur	211
--	-----

Chapitre 6 **Les décorateurs**

1. Introduction	213
2. Décorateur de classe	215
3. Décorateur de méthode	217
4. Décorateur de propriété	220
5. Décorateur de paramètre	225
6. Fabrique de décorateurs	227
7. Métadonnées.	229

Chapitre 7 **Asynchronisme**

1. Introduction	239
2. Callback hell	240
3. Promesse	242
3.1 Historique	242
3.2 Promesses ECMAScript 2015.	243
3.2.1 Résolution des promesses.	243
3.2.2 Then, catch et finally	245
3.2.3 Promise.all et Promise.race.	248
3.2.4 Conclusion	249
4. Async/await	250

Chapitre 8

Système de types avancés

1. Introduction	255
2. Alias de type	255
3. Type never	259
4. Type union et intersection	261
4.1 Union	261
4.2 Intersection	263
5. Type guards	266
5.1 Introduction	266
5.2 Opérateur typeof	266
5.3 Opérateur instanceof	268
5.4 Opérateur in	270
5.5 Union discriminante	271
5.6 Type guards définis par l'utilisateur	273
5.7 Gestion de null et undefined	276
5.8 Cas impossible	279
6. Types littéraux	280
7. Type index	282
8. Mapped type	285
8.1 Avant TypeScript 2.1	285
8.2 Les opérateurs	287
8.3 Mapped type natif	289
9. Affirmation constante	292
10. Type conditionnel	294
10.1 Les bases	294
10.2 Distributivité	296
10.3 Opérateur infer	298
10.4 Type conditionnel natif	299

Chapitre 9
TypeScript et la programmation fonctionnelle

- 1. Introduction 303
- 2. Fonction pure et impure. 306
- 3. Immutabilité. 309
- 4. Itération. 316
- 5. Conditions. 320
- 6. Fonction partielle 321
- 7. Currying 322
- 8. Pattern Matching 324
- 9. Composition de fonctions 328
- 10. Pour aller plus loin... 333

Chapitre 10
Un premier projet avec Node.js

- 1. Introduction 335
- 2. Préparation de la base de données 338
- 3. Mise en place du projet 342
 - 3.1 Création du fichier package.json 343
 - 3.2 Mise en place de TypeScript 344
 - 3.2.1 Installation 344
 - 3.2.2 Configuration de TypeScript 345
 - 3.3 Amélioration de l'environnement de développement 349
 - 3.3.1 Exécution de l'application 349
 - 3.3.2 Débogage 351
 - 3.4 Une première application. 353
 - 3.4.1 Mise en place de la bibliothèque Express.js. 353
 - 3.4.2 Gestion des variables d'environnement. 356
 - 3.4.3 Une première requête SQL. 357

4.	Création du Framework MVC	359
4.1	Registre des routes	360
4.2	Décorateurs de routes	366
4.2.1	Le décorateur Controller	366
4.2.2	Le décorateur Route	368
4.3	Mise en œuvre du Framework MVC	370
4.3.1	Création du contrôleur	370
4.3.2	Création des routes	372
4.4	Middleware Express.js	374
5.	Accès aux données	377
5.1	Entité métier	378
5.2	Dépôt de données abstrait	379
5.2.1	Dépôt	379
5.2.2	Type d'opération	380
5.2.3	Dépôt de données abstrait	380
5.3	Définition des modèles métiers	388
5.4	Un premier dépôt de données : EmployeeRepository	389
5.5	Utilisation du dépôt de données EmployeeRepository	391
6.	Injection des dépendances	393
6.1	Service de gestion des dépendances	393
6.2	Registre de clé	396
6.3	Décorateurs de gestion des dépendances	397
6.4	Résolution des dépendances lors de la création des contrôleurs	401
6.5	Enregistrement d'une dépendance	403
7.	Finalisation de l'API	405
7.1	Ajout d'un employé	405
7.2	Gestion des équipes	414
7.3	Changer l'équipe d'un employé	420
8.	Conclusion de l'exercice	426
	Index	427

Avant-propos

1. Pourquoi cet ouvrage ?	13
2. À qui s'adresse cet ouvrage ?	14
3. Structure de l'ouvrage	14

Chapitre 1 Introduction

1. État des lieux du développement web	17
2. Angular	18
2.1 Organisation par composants	18
2.2 TypeScript	19
2.3 Les spécifications ES6	19
2.4 DOM Virtuel	19
2.5 Rendu côté serveur possible	20
3. AngularJS vs Angular	21

Chapitre 2 Ma première application

1. Introduction	23
2. Le setup	23
2.1 Installation de npm	23
2.2 Installation d'Angular CLI et initialisation du projet	25
3. Mon premier composant	27
4. Mon premier module	28
5. Lancement de l'application	30

Chapitre 3

Fondamentaux d'Angular

1. Introduction	33
2. Les composants	33
3. Les templates	35
3.1 Binding	36
3.2 Événements JavaScript	38
3.3 Listes	39
3.4 Afficher/cacher un élément	41
3.5 Liens	44
3.6 Styles	45
3.7 Directives	49
4. Les décorateurs	51
5. Les pipes	52
5.1 Utiliser un pipe	52
5.2 Les pipes du framework	54
5.3 Créer un pipe	55
5.3.1 Les pipes de transformation	55
5.3.2 Les pipes filtres	57
5.4 Les pipes purs et impurs	58
6. Les modules	60
7. Observable et promise	61
8. Préparer l'application pour la production	63

Chapitre 4
TypeScript

- 1. JavaScript 65
- 2. TypeScript..... 67
 - 2.1 Syntaxe 68
 - 2.1.1 Variables 68
 - 2.1.2 Fonctions..... 69
 - 2.1.3 Classes..... 69
 - 2.1.4 Arrow Function 71
 - 2.1.5 Union type 72
 - 2.1.6 Interface 73
 - 2.1.7 Générique 74
 - 2.1.8 Les décorateurs 74
 - 2.2 Typings 75
 - 2.3 Compilation 76
 - 2.4 EcmaScript 6 et 7 dès maintenant..... 77
 - 2.5 Angular et TypeScript 78

Chapitre 5
Angular CLI

- 1. Introduction 79
- 2. Qu'est-ce qu'Angular CLI ? 79
 - 2.1 La définition 79
 - 2.2 Les commandes principales 81
 - 2.2.1 Créer un nouveau projet Angular CLI : ng new 81
 - 2.2.2 Compiler l'application : ng build 82
 - 2.2.3 Lancer l'application avec ng serve 85
 - 2.2.4 Exécuter les tests avec Angular CLI..... 86
 - 2.3 Les commandes de scaffolding..... 90
 - 2.4 La génération d'un composant..... 91

3.	Configurer un projet Angular CLI	97
3.1	Schéma du fichier de configuration d'Angular CLI	97
3.2	Gérer les assets lors de la compilation	102
3.3	Intégrer une librairie externe	104
3.4	Intégrer un préprocesseur CSS	106
3.5	Ajouter des entités de manière globale	106
3.5.1	Ajouter un script globalement	106
3.5.2	Ajouter une feuille de style globalement	107
3.6	Éjecter la configuration Webpack	110
4.	Compilation Ahead-Of-Time (AOT)	110

Chapitre 6 Les composants

1.	Introduction	115
2.	Qu'est-ce qu'un composant ?	115
2.1	Une première définition	115
2.2	Créer un premier composant	117
2.2.1	Syntaxe inline	117
2.2.2	Une réelle syntaxe, découpée en plusieurs fichiers	118
2.2.3	Utiliser un composant dans son application	119
2.3	Afficher les propriétés d'un composant	121
3.	Les inputs et outputs	123
3.1	Les inputs d'un composant	123
3.1.1	Déclarer une variable en tant qu'Input	123
3.1.2	Un exemple concret avec une liste	124
3.1.3	Donner un nom personnalisé à son input	126
3.2	Les outputs d'un composant	127
3.3	Donner un nom personnalisé à son output	130
4.	Interaction entre composants	130
4.1	Passer une donnée du parent vers l'enfant à l'aide d'un input	131
4.2	Intercepter un changement de valeur à l'aide d'un setter	133

- 4.3 Notifier le parent à l'aide d'un EventEmitter en output 135
- 4.4 Observer les changements d'input avec ngOnChanges 137
- 4.5 Utiliser une variable locale 140
- 5. Les décorateurs @ViewChild et @ViewChildren 142
 - 5.1 Récupérer une référence avec @ViewChild 142
 - 5.2 Récupérer plusieurs références avec @ViewChildren 145
- 6. Les composants Angular et la View Encapsulation 147
 - 6.1 Le Shadow DOM 147
 - 6.2 Spécifier une View Encapsulation 149
 - 6.3 Les types de View Encapsulation 149
 - 6.3.1 La View Encapsulation émulée 150
 - 6.3.2 La View Encapsulation native 152
 - 6.3.3 Aucune View Encapsulation 153

Chapitre 7
Les services

- 1. Introduction 157
- 2. Qu'est-ce qu'un service ? 157
- 3. Déclarer son service 158
- 4. Utiliser son service 158
- 5. Rendre son service asynchrone 161
 - 5.1 Les promesses 161
 - 5.2 Les observables 164
 - 5.3 Que choisir ? 170
- 6. Notifier lorsque les données changent 170

Chapitre 8

L'injection de dépendances

1. Principe de base.	175
2. Injection de dépendances dans Angular	176
2.1 Enregistrement global	177
2.2 Enregistrement dans un composant	178
2.3 Injecter une dépendance	179
3. Provider	180
3.1 useClass	180
3.2 UseExisting	181
3.3 UseFactory	183
3.4 UseValue	186
3.5 OpaqueToken	186
3.6 Dépendance optionnelle.	190
3.7 Injection restreinte	191
3.8 Restriction de l'enregistrement d'une dépendance.	193

Chapitre 9

Le requêtage HTTP

1. Introduction	195
2. Obtenir et envoyer des données	195
3. Transformer des données	198
4. Communiquer de manière sécurisée	199
5. Simuler le requêtage HTTP	202

Chapitre 10
Les interactions utilisateur

- 1. Qu'est-ce que l'event binding ? 205
- 2. S'abonner à un événement 206
- 3. Récupérer une entrée utilisateur 207
 - 3.1 Comment manipuler l'objet \$event ? 207
 - 3.2 Utiliser un typage fort pour \$event. 208
 - 3.3 Une alternative grâce à un template reference variable 209
 - 3.4 Utiliser un template reference variable avec un événement .. 210
 - 3.5 Filtrer les entrées utilisateur. 211

Chapitre 11
Les formulaires

- 1. Les formulaires basés sur un template 215
- 2. Créer un composant formulaire. 216
 - 2.1 Le composant 216
 - 2.2 La vue et le data binding 217
 - 2.2.1 La syntaxe ngModel 218
 - 2.2.2 ngModel en détail. 218
 - 2.2.3 L'utilisation de ngModel dans un cas concret 218
 - 2.3 Intégrer le formulaire dans l'application 219
- 3. Les états et la validité d'un champ 220
 - 3.1 Les états d'un input 220
 - 3.2 Styliser selon la validité 222
- 4. Soumettre le formulaire 223
- 5. Les formulaires et les FormControls 224
 - 5.1 Les contrôles et les groupes de contrôles. 224
 - 5.2 Les validateurs intégrés 227
 - 5.3 Créer un validateur personnalisé 228
 - 5.4 Les validateurs asynchrones. 229

Chapitre 12

Le routage

1. Introduction	231
2. Définir les routes d'une application	231
3. Le rendu de composant	233
4. Naviguer dans son application.	235
5. Récupération des données de routage	236
6. Outlet nommé	240
6.1 Définir des outlets nommés.	240
6.2 Naviguer avec des outlets nommés	242
7. Resolver	243

Chapitre 13

Les directives

1. Introduction	249
2. Qu'est-ce qu'une directive ?	249
2.1 Introduction	249
2.2 Directives communes	250
2.2.1 ngIf	250
2.2.2 ngFor	250
2.2.3 ngStyle.	251
2.2.4 ngClass	251
3. Les directives d'attribut	252
3.1 Créer une directive d'attribut	252
3.2 Interagir avec les événements du DOM	254
3.3 Passer des valeurs aux directives d'attribut.	255
4. Les directives structurelles	258
4.1 La balise <template> et l'astérisque	258
4.2 Créer une directive structurelle	259

Chapitre 14

Tester son application

1. Introduction	265
2. Les tests unitaires	265
2.1 Introduction aux tests avec Jasmine	266
2.2 Exécuter du code avant ou après chaque test	269
2.3 Les matchers mis à disposition	270
2.3.1 Comment utiliser un matcher ?	271
2.3.2 Des exemples de matchers	271
2.3.3 Négation d'un matcher	274
2.4 Créer un matcher personnalisé	275
2.4.1 Créer une librairie de matchers personnalisés	275
2.4.2 Utiliser un matcher personnalisé	277
2.5 Les composants	278
2.5.1 Le composant à tester	278
2.5.2 Le TestBed	279
2.5.3 Vérifier que le composant est bien instancié	280
2.5.4 Contrôler les propriétés du composant	281
2.5.5 S'assurer que le rendu du composant soit cohérent	281
2.6 Les services	283
2.6.1 Le service à tester	283
2.6.2 Tester le service	283
2.6.3 S'assurer que le service est bien injecté	284
2.7 Les directives	285
2.7.1 La directive à tester	285
2.7.2 Tester la directive	286
2.8 Injecter un faux service	288
3. Les tests e2e	289
3.1 Lancer les tests e2e	290
3.2 Écrire un test e2e	291
3.2.1 Interagir avec le navigateur	292
3.2.2 Récupérer un élément du DOM et interagir avec	293

3.2.3 Interagir avec les éléments DOM	294
3.2.4 Tester le composant	296
3.3 Écrire un test e2e, en mieux	297

Chapitre 15

Le cross-platform avec Angular

1. Apache Cordova	301
1.1 Créer un projet Apache Cordova	302
1.2 Angular dans Apache Cordova	304
1.3 Utiliser un plug-in Apache Cordova	309
2. Ionic 2	314
2.1 Créer un projet Ionic	314
2.2 Utiliser les composants Ionic	315
2.3 Utiliser Ionic DevApp	317
2.4 Utiliser un plug-in avec Ionic	318

Chapitre 16

Pour aller plus loin

1. Introduction	321
2. Rendu côté serveur	321
2.1 Principe de la mise en place	322
2.2 Ajouter le rendu côté serveur	323
3. La détection de changement	333
3.1 Pourquoi la détection de changement ?	333
3.2 Les Zones, ou comment notifier Angular ?	336
3.2.1 Le comportement de la détection de changement	338
3.2.2 L'immutabilité et la stratégie de détection de changement OnPush	339
3.3 Encore plus de contrôle sur la détection de changement	342

- 4. Le cycle de vie d'un composant 343
 - 4.1 La présentation des lifecycle hooks 343
 - 4.1.1 Les différents hooks 343
 - 4.1.2 Utiliser un lifecycle hook 344
 - 4.2 Le cycle de vie d'un composant 345
 - 4.2.1 Le constructeur 348
 - 4.2.2 `ngOnInit` 348
 - 4.2.3 `ngOnChanges` 349
 - 4.2.4 `ngDoCheck` 350
 - 4.2.5 `ngAfterContentInit` 350
 - 4.2.6 `ngAfterContentChecked` 352
 - 4.2.7 `ngAfterViewInit` 352
 - 4.2.8 `ngAfterViewChecked` 353
 - 4.2.9 `ngOnDestroy` 353

- Index 355