- 1. Eclipse の起動
- (1) Finder の「移動」「アプリケーション」の順にクリックしアプリケーションのウインドウを開き、「Eclipse_4.7.0」のアイコンをダブルクリックします.



(2) macOS Sierra では、はじめて Eclipse_4.7.0 を起動する時に、以下の表示が出ます.
 「OK」をクリックします。2回目以降の起動時には、表示されません。



(3) 「control」キーを押しながら、「Eclipse_4.7.0」のアイコンをクリックし、「開く」 をクリックします. この操作は、2回目以降の起動時には、必要ありません.

•			🛃 アプリケーション	-	
×		Š	() 開く () () () () () () () () () () () () ()		
Adobe Reader	App Store	Aut	ゴミ箱に入れる		.7.0
FaceTime	Firefox	For	情報を見る 名前を変更 "Eclipse_4.7.0"を圧縮 複製 エイリアスを作成 "Eclipse_4.7.0"をクイックルック 共有		ithor
			"Eclipse_4.7.0"をコピー		
iMovie	iMovie 9.0.4	iF	表示オプションを表示		oft
			タグ		cator
x∎	88	N	•••••		
			シマンテック: 選択したファイルを; Finderに表示	スキャンして脅威を調べる	

(4) 以下の表示が出る場合は、「開く」をクリックします. 2回目以降の起動時には、表示されません.



(5) 「この選択をデフォルトとして使用し、今後この質問を表示しない(U)」をチェック し、「OK」をクリックします.この操作を行うと、2回目以降の起動時には、表示 されません.

00	Eclipse ラン	ノチャー	
ワークスペースとしての Eclipse は設定や開発成果を	ディレクトリー選択 匆を格納するためにワークスペース・テ	ディレクトリーを使用します。	
ワークスペース:/works	pace	<u> </u>	参照
 ごの選択をデフォルトと 最近使ったワークスペース 	して使用し、今後この質問を表示しな	L)	
		キャンセル	起動

(6) しばらくすると、以下のウインドウが表示され、起動が完了します.

	• • • • • • • • • • • • • • • • •	× 0 ·						E 0000++
						- 0	PE 791 23 1 93.9	· 7-7 · 0
	1							- P - P
							表示するアウトラインはあり	ません。
	🔛 RM 🔛 🚑 93.9 🕒 327-1/ 🛅 70/0	F-4-						> > > = a
	0 項目 		and the					
	ज्ञातन <u>ा</u>	~ yy-x	A.A.	67-939	SH .			
UTF-8 LF 0項目が漏死されました						. i e	SEUUINT .	- 0

- 2. プログラムの入力
- (1) 最初に C プロジェクトを新規に作成します.「新規」「C プロジェクト」の順にクリ ックします.

	😑 😑 🔵 /Applicat	tions/Eclipse_4.7.0.app/Conte
<	 ○ ○<th>• 🕸 • 🕢 • 🚱 • 💁 • 🚑 🖄</th>	• 🕸 • 🕢 • 🚱 • 💁 • 🚑 🖄

(2) この例では、プロジェクト名を「hello」と入力した後、「MacOSX GCC」を選択し、
 「完了」をクリックします.

• • •	C プロジェクト	
C プロジェクト 選択したタイプの C プロ:	ェクトを作成します	2
プロジェクト れ : hello		
🗹 デフォルト・ロケーシ	ンを使用	
ロケーション: /Applica	ns/Eclipse_4.7.0.app/Contents/workspace/hello 参照	
ファイル	>ステムを選択: デフォルト ◇	
プロジェクト・タイプ:	ツールチェーン:	
 GNU Autotools static ライブラリー 共用ライブラリー 実行可能 空のプロジェク Hello World J Makefile プロジェク 	Cross GCC MacOSX GCC ISI C プロジェクト 7ト	
💟 プラットフォームでサ	ートされる場合のみ、プロジェクト・タイプとツールチェーンを表示	
0		

(3) プロジェクト・エクスプローラに「hello」が表示されます. 今後は、プログラム毎にプロジェクトを新規に作成していきます.

		· @ . # . 0.0	p/Contents/workspa	ce - Eclipse	. A. O. C		
					0.40	5.75H2	Ebeica
プロジェクト・エクスプローラ 😚 😐 🗖						D= 7 52 10 2	», 🗆 j
					U	01 · 00 [0] ·	
▶ 😂 hello						表示するアウトライン	はありません。
		24 D 22V-k T	2/17-4				~
	 	ネタ 🕒 コンソール 🔳 プレ	21177-4-				⊽ □ (
	 ○ 時間 23 - 9 / 9 / 9 / 9 / 9 / 9 / 9 / 9 / 9 / 9	ネタ 🕒 コンソール 🔲 プレ	۲/1₹ ∧ ۲۷−3	152	ענילים	* *	▽ □
	 ● 開闢 23 (2) 570 ○ 項目 取用 	スタ 🕒 コンソール 🔲 プロ	ロバティー ヘ リソース	152	ロケーション	* *	~ ~ (
	 ・ 問題 23 ・ ダン ・ ダン ・ の項目 説明 ・ ・ ・	20 🔲 JU-JL 🔄 JE	2707-4- ▲ IJY-ス	172	ロケーション	*	V = (
	 () 開闢 (2) 97 () 項目 取明 	29 🖸 379-J. 🔲 H	2/(₹-4- > ⊻У-2	152	ロケーション	⇒ ÷	~ ~ (
	前編 33 9 9 7 0 項目 取列	スク 🖸 コンソール 🔲 プロ	2//₹ > УУ-2	152	ロケーション	⇒ 0 ×	~

(4) 次はソースファイルを作成します.プロジェクト・エクスプローラの「hello」が選 択されている状態で、「新規」「ソース・ファイル」の順にクリックします.



(5) この例では、ソース・ファイル名は「hello.c」と入力し、「完了」をクリックします.
 C言語のソースファイルの拡張子は、「.c」になります. 拡張子を忘れないようにしてください.

• •	新規ソース・ファイ	JL	
ソース・ファイル 新しいソース・ファ・	イルを作成します。		C
ソース・フォルダー:	hello		参照
ソース・ファイル テンプレート:	hello.c デフォルト C ソース・テンプレート		構成
?		キャンセル	完了

(6) プロジェクト・エクスプローラには、hello プロジェクトの配下に、「インクルード」
 と「hello.c」が表示されます.エディタには、自動生成されたコードが表示されます.



(7) エディタの 9 行目から、以下のプログラムを入力し、「保管」をクリックします. MacOS の場合は、¥の入力に注意が必要です.「option」キーを押しながら「¥」を 入力してください.新規にソースファイルを作成した場合は、「保管」を忘れないよ うにしてください.なお、「(」や「<」を入力すると、自動的に「)」と「>」も入力 されます.

#include <stdio.h>

int main(void)

{

printf("Hello World¥n");

return 0;





- 3. ビルド
- (1) 「すべてビルド」をクリックします.



(2) ビルドの結果がコンソールに表示されます.

		EX REC/C++
と プロジェクト・エクスプローラ ☆ □ □	c helio.c 🕄	 1 🗐 🤊 🐂 🗖 🗖
 ● ない ● はい 	100 /* hells.c. * Created on: 2017/00/06 * Arbon: 2007 * FineInde Catalo. No. 101 in temial(void) 101 in temial(void) 101 in temial(void) 101 in temial(void)	a) t ⁴ 2

「保管」を忘れて「すべてビルド」を行うと、コンソールにエラーが表示されます.

			イック・アクセス E EC/C++
と プロジェクト・エクスプローラ ☆ □ □	e hello.c 😫		1 🚼 7 🕱 🗐 🤊 "1 🗖 🗖
E Shelio ► Shelio ► Shelio ► Shelio.c	<pre></pre>		P ⊟ I ² ₂ ⊗ k ² ∘ # S stoic.h • main(void) : int
			- R & - D. P (
(CDT Build Console [hello]		
	building taract halls hunding: MeG 5 Lukar pcoor hallo', Abilo 5 Lukar pcoor hallo', Abilo 5 Lukar (Abilo 5 Lukar) (Abilo 5 Lukar)	(use -v to see invocation)	
	A pression pression of the	22	

4. 実行

「実行」をクリックします.

• • •	a hello/hello.c - /Applications/Eclipse_4.7.0.app/Contents/w
📑 • 📰 🕼 🛞 • 🗞 • 🔜 🕥 • 💷 🔌	
┣ プロジェクト・エクスプローラ 🔉 🖓 🗖	le hello.c S
E hello ■ ゴンクルード ■ ■	2 * hello.c 3 * 4 * Created on: 2017/09/05 5 * Author: dohi 6 */
▶ 💼 hello.c	9 #include <stdio.h></stdio.h>
	13 * printf("Hello World¥n"):↓ 14]1 15

(2) 「C/C++アプリケーション」をダブルクリックします. この表示は,新規にソース ファイルを作成した時に出ます.

	実行構成	
構成の作成、管理、および実行		
 マィルター入力 マィルター入力 C C/C++ アブリケーション C C/C++	このダイアログから起動設定を構成します: □ - 選択したタイプの構成を作成するには、「新規」ボタンを押します。 □ - 選択した構成をコピーするには、「複製」ボタンを押します。 → - 選択した構成を除去するには、「削除」ボタンを押します。 → - フィルター・オプションを構成するには、「フィルター」ボタンを押します。 - 既存構成を選択して、それを編集または表示します。 <u>パースペクティブ</u> 設定ページから、起動パースペクティブの設定を構成します。	
フィルター一致: 4 / 4 項目		
?		開じる実行

	実行構成	
構成の作成、管理、および実行		
· • × □ × 1	名前: hello Debug	
フィルター入力	(×1>) (×)= 引数 ■ 環境 □ 共通	
▼ C C/C++ アプリケーション C hello Debug	プロジェクト:	
C C/C++ リモート・アプリケー	hello	参照
記載がループ	C/C++ アプリケーション:	
▶ 短期シルーン (非推奨)	Debug/hello	
	変数 プロジェクトの検索. 起動前に (必要に応じて) ビルド	
	ビルド構成: Select Automatically	0

	ビルド構成: Select Automatically	
	○ 自動ビルドを使用可能にする	○ 自動ビルドを使用不可にします
	○ ワークスペース設定の使用	ワークスペース設定の構成
		前回保管した状態に戻す 適用
ィルター一致: 5 / 5 項目		
0		1911 T
0		間じる 美行

(4) 実行結果がコンソールに表示されます.

プロジェクト・エクスプローラ 52 日日	E bella c 12		9= 7 m = 4 ". = F
Image: Second Secon	Leg helio.c 23		E 12 V 1 C
	● R目 ● タスク ● コンソール 23 ● プロパティー <株了> (exit value: 0) helio Debug (C/C++ アプリケーション) /Applica Hello Morld	I X 🔌 🖟 🖬 🕼 🧔	ice/helio/Debug/helio (2017/09/

実行結果の表示に¥n が表示されている場合は、¥の入力時に「option」キーを押しながら 「¥」を入力していないことになります.「¥」を削除し、「option」キーを押しながら「¥」 を入力し直してください.

to deal and the termination of the second se		
Tロジェクト・エクスプローラ 22 □ □ Comparison	C helioc 3	E7X V * = C Stdio.h o main(void) : int
	Pla の タスク ロコンソーレ 23 ロ ブロバティー 「茶 後 し 回 の Contents/works KT> (exit value: 0) helio Debus [C/C++ アブリケーション] /Applications/Eclipse_4.7.0.app/Contents/works Hello World¥n	pace/hello/Debug/hello (2017/09/1

- 5. プログラムを書き換えた後のビルド
- (1) 誤り等を修正した場合や、ソースファイルのプログラムを書き換えた場合は、「保管」 「すべてビルド」の手順に従います.しかし、操作が煩雑になるので、「実行」をク リックすると良いでしょう.



(2) 「常に起動前にリソースを保管する」をチェックし、「OK」をクリックします.

- 保管して起動
保管するリソースを選択してください:
hello.c
すべて選択 選択をすべて解除
✔ まに起動前にリソースを保管する
(?) ++>+

(3) エラーが無ければ「保管」「すべてビルド」「実行」が自動的に行われます.エラー がある場合は、「保管」「すべてビルド」まで行われます.エラーを修正し、「実行」 を行います.今後は、エラーの原因を早く見つけられるようになることが重要です.