**1.**

What is a **Theme** composed of?

**1 / 1 point**



A collection of views



A collection of attributes



A collection of composables



**Correct**

Correct! **Theme** is represented by attributes such as height and color.

**2.**

.Which color is represented by the following hexadecimal value?

**0xFF000000**

**1 / 1 point**



white



gray



black



blue



**Correct**

Correct! This hexadecimal value represents a black color. The 0 means that it is a minimum value for each color, so **000000** means black. However, the first two **FF** characters represent the alpha channel where the maximum value is required to make the color fully opaque.

**3.**

Which of the following correctly creates a text composable with bold text?

**1 / 1 point**



1

2

3

4

5

6

Text(

    text = “Hello World!”,

    fontSize = 30.sp,

    font = Bold

)



Text( text = “Hello World!”, fontSize = 30.sp, font = Bold )



1

2

3

4

5

Text(

    fontSize = 30.sp,

    fontWeight = Bold

)



Text( fontSize = 30.sp, fontWeight = Bold )



1

2

3

4

5

6

Text(

    text = “Hello World!”,

    fontSize = 30.sp,

    fontWeight = Bold

)



Text( text = “Hello World!”, fontSize = 30.sp, fontWeight = Bold )



1

2

3

4

5

6

Text(

    text = “Hello World!”,

    fontSize = 30.sp,

    weight = Bold

)



Text( text = “Hello World!”, fontSize = 30.sp, weight = Bold )



Correct! A Text composable accepts an optional parameter of **fontWeight** for styling the text.

**4.**

Which of the following correctly creates a text composable with underlined text?

**1 / 1 point**



4

1

2

3

Text(

    style = TextDecoration.Underline

)



Text( style = TextDecoration.Underline )



**Text(**

1

2

3

4

5

6

7

    style = TextStyle(

        text = “Hello World!”,

        fontSize = 25.sp,

        textDecoration = underline

    )

)



style = TextStyle( text = “Hello World!”, fontSize = 25.sp, textDecoration = underline ) )



1

2

3

4

5

6

7

8

Text(

    style = TextStyle(

        text = “Hello World!”,

        fontSize = 25.sp,

        textDecoration = TextDecoration.Underline

    )

)



Text( style = TextStyle( text = “Hello World!”, fontSize = 25.sp, textDecoration = TextDecoration.Underline ) )



1

2

3

4

5

6

7

8

Text(

    style = TextStyle(

        text = “Hello World!”,

        fontSize = 25.sp,

        decoration = TextDecoration.Underline

 )

)



Text( style = TextStyle( text = “Hello World!”, fontSize = 25.sp, decoration = TextDecoration.Underline ) )



Correct! The text-decoration property can be used to apply underline text. The **TextDecoration** combine function enables the application of multiple decorations.

**5.**

Which of the following correctly adds **Text** composable as the content for a **Surface**?



6

7

8

9

            )

}

)



Surface( { Text( text = “Hello World”, fontSize = 30.sp ) } )



1

2

3

4

5

6

7

8

9

10

Surface(

    modifier = Modifier

        .fillMaxSize()

) {

    Text(

            text = “Hello World”,

            fontSize = 30.sp

            )

}



Surface( modifier = Modifier .fillMaxSize() ) { Text( text = “Hello World”, fontSize = 30.sp ) }



1

2

3

4

5

6

7

8

Surface(

    modifier = Modifier

        .fillMaxSize()

    text = “Hello World”,

    fontSize = 30.sp

)

}



Surface( modifier = Modifier .fillMaxSize() text = “Hello World”, fontSize = 30.sp ) }



1

2

3

4

5

6

7

8

9

10

Text(

    modifier = Modifier

        .fillMaxSize()

) {

    Surface(

            text = “Hello World”,

            fontSize = 30.sp

            )

}



Text( modifier = Modifier .fillMaxSize() ) { Surface( text = “Hello World”, fontSize = 30.sp ) }



**6.**

Which of the following is an optional modifier that can be used to change the color of text within a **Text** composable?

**1 / 1 point**



**text**



**color**



**textColor**



**fontColor**



**Correct**

Correct! A Text composable accepts over 15 optional parameters including **color** for styling the text.

**7.**

Imagine you have a **Surface** with a **Text** composable as its content. Which of the following modifiers will set the minimum height required to show the text?

**1 / 1 point**



**padding**



**wrapContentSize**



**fillMaxSize**



**align**



**Correct**

Correct! It is a good practice to set the minimum height required to show text.

**8.**

Which of the following is the composable that animates the appearance and disappearance of its content?

**1 / 1 point**



**Animation**



**AnimatedVisibility**



**VisibleAnimation**



**Visibility**



**Correct**

Correct! The **AnimatedVisibility** composable smooths out **Text** transition from one state to another, from the component being visible to invisible and vice versa.

**9.**

Which of the following will make the text “Hello World” slide up to disappear and slide down to disappear?

**0 / 1 point**



1

2

3

4

5

6

7

8

9

10

11

12

13

var visible by remember {

    mutableStateOf(true)

}

Column {

    if  (visible)  {

        Text(text = “Hello World”)

    }

    Button(onClick = {  visible = !visible  })  {

        Text(“Button“)

}

}



var visible by remember { mutableStateOf(true) } Column { if (visible) { Text(text = “Hello World”) } Button(onClick = { visible = !visible }) { Text(“Button“)



2

3

4

5

6

7

8

9

10

11

12

1

    mutableStateOf(true)

}

Column  {

    AnimatedVisibility(visible)  {

        Text(text = “Hello World”)

    }

    Button(onClick = {  visible = !visible  })  {

        Text(“Button  “)

    }

}

var visible by remember {



var visible by remember { mutableStateOf(true) } Column { AnimatedVisibility(visible) { Text(text = “Hello World”) } Button(onClick = { visible = !visible }) { Text(“Button “) }



1

2

3

4

5

6

7

8

9

var visible by remember {

    mutableStateOf(true)

}

Column {

    if  (visible)  {

        Text(text = “Hello World”)

    }

    Button(onClick = Text(“Button  “))



var visible by remember { mutableStateOf(true) } Column { if (visible) { Text(text = “Hello World”) } Button(onClick = Text(“Button “))



1

2

3

4

5

6

7

8

9

10

var visible by remember {

    mutableStateOf(true)

}

Column  {

    AnimatedVisibility(visible) {

        Text(text = “Hello World”)

    }

    Button(onClick = Text(“Button“))

}



var visible by remember { mutableStateOf(true) } Column { AnimatedVisibility(visible) { Text(text = “Hello World”) } Button(onClick = Text(“Button“)) }



**10.**

Which of the following parameters is used to control how long a given transition will last?



**exit**



**AnimatedVisibilty**



**enter**



**animationSpec**



**Correct**

Correct! The animation duration is provided in milliseconds.