



PROGRAMOWANIE URZĄDZEŃ MOBILNYCH

WYKŁAD 6 Jetpack Compose

- LazyColumn, LazyRow
- Listy
- Grid
- Pager
- State

Jetpack Compose



Column

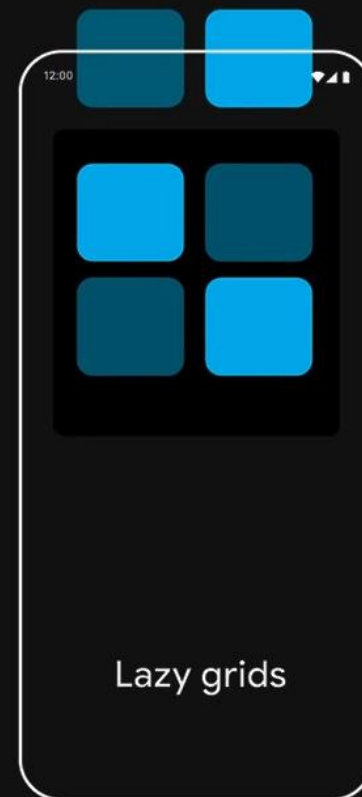
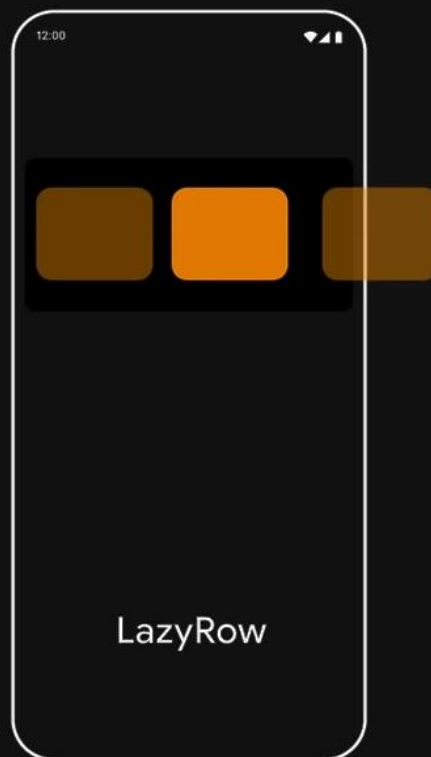
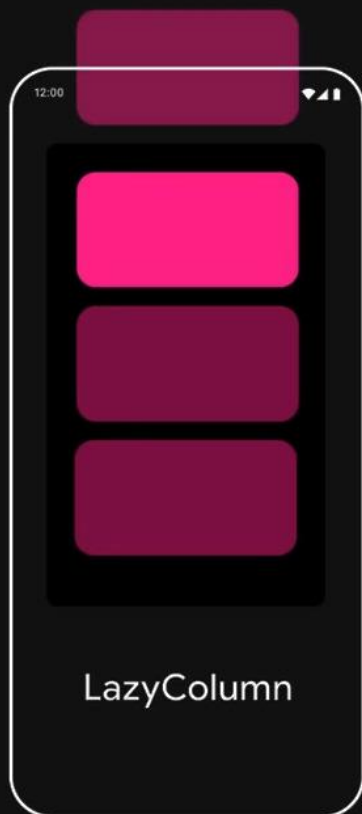


Row



Box

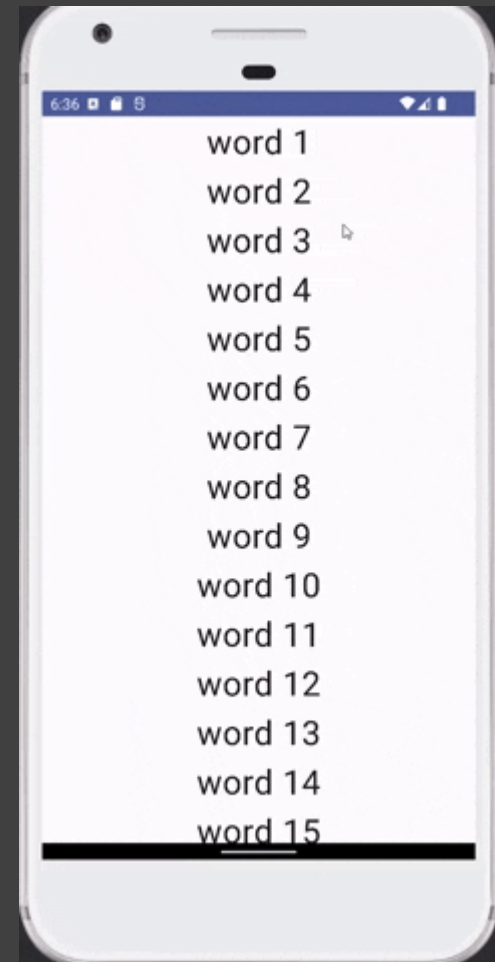
Jetpack Compose



LazyColumn

```
private fun generateWordList(): MutableList<String> {  
    return (1..50).map { "word $it" }.toMutableList()  
}
```

```
@Composable  
fun ListOfWords(words: MutableList<String>){  
    LazyColumn{  
        items(words.size){  
            var word by remember {  
                mutableStateOf(words[it])  
            }  
            Text(  
                text = word,  
                fontSize = 32.sp,  
                textAlign = TextAlign.Center,  
                modifier = Modifier  
                    .fillMaxWidth()  
                    .padding(2.dp)  
                    .clickable {  
                        word += "Clicked!!!"  
                        words[it] = word  
                    }  
            )  
        }  
    }  
}
```



```
LazyColumn {  
    // Add a single item  
    item {  
        Text(text = "First item")  
    }  
  
    // Add 5 items  
    items(5) { index ->  
        Text(text = "Item: $index")  
    }  
  
    // Add another single item  
    item {  
        Text(text = "Last item")  
    }  
}
```

```
LazyColumn {  
    // Add a single item  
    item {  
        Text(text = "First item")  
    }  
  
    // Add 5 items  
    items(5) { index ->  
        Text(text = "Item: $index")  
    }  
  
    // Add another single item  
    item {  
        Text(text = "Last item")  
    }  
}
```

```
/**  
 * import androidx.compose.foundation.lazy.items  
 */  
LazyColumn {  
    items(messages) { message ->  
        MessageRow(message)  
    }  
}
```

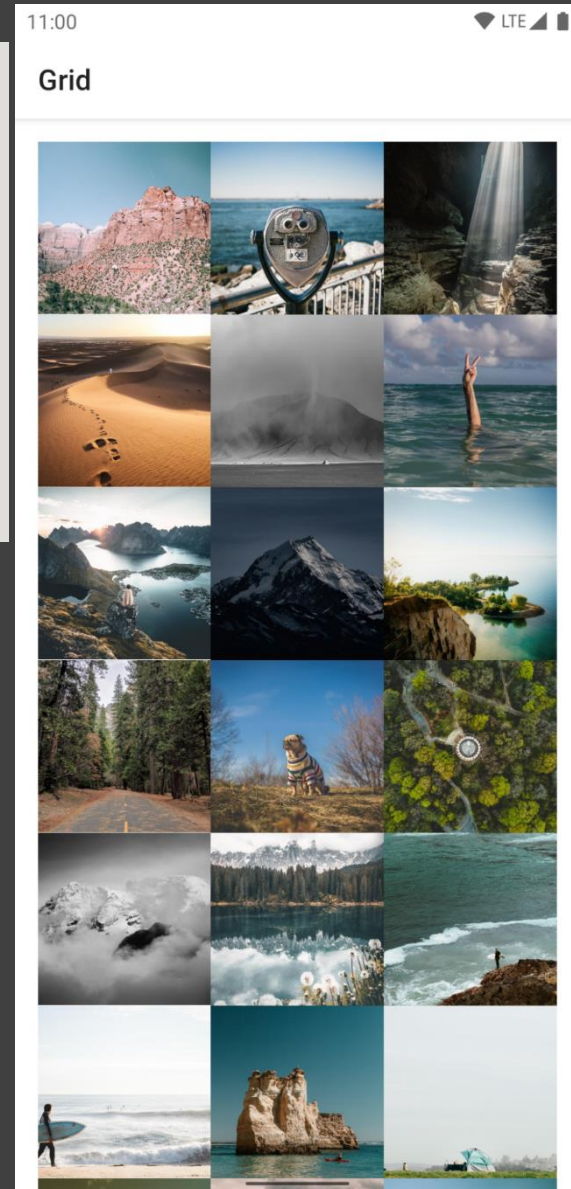
```
import androidx.compose.foundation.border
import androidx.compose.foundation.layout.height
import androidx.compose.foundation.layout.wrapContentSize
import androidx.compose.foundation.lazy.grid.LazyVerticalGrid
import androidx.compose.foundation.lazy.grid.items
import androidx.compose.foundation.lazy.grid.itemsIndexed
import androidx.compose.material.Text

val itemList = (0..5).toList()
val itemsIndexedList = listOf("A", "B", "C")

val itemModifier = Modifier.border(1.dp, Color.Blue).height(80.dp).wrapContentSize()

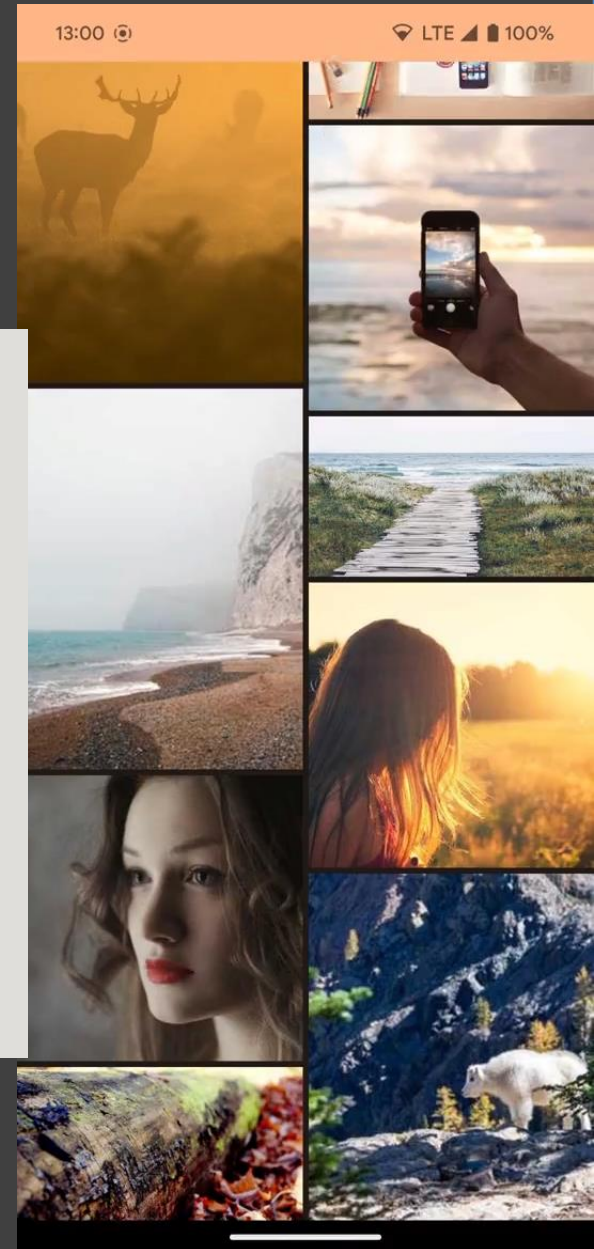
LazyVerticalGrid(
    columns = GridCells.Fixed(3)
) {
    items(itemList) {
        Text("Item is $it", itemModifier)
    }
    item {
        Text("Single item", itemModifier)
    }
    itemsIndexed(itemsIndexedList) { index, item ->
        Text("Item at index $index is $item", itemModifier)
    }
}
```

```
LazyVerticalGrid(  
    columns = GridCells.Adaptive(minSize = 128.dp)  
) {  
    items(photos) { photo ->  
        PhotoItem(photo)  
    }  
}
```



LazyVerticalStaggeredGrid

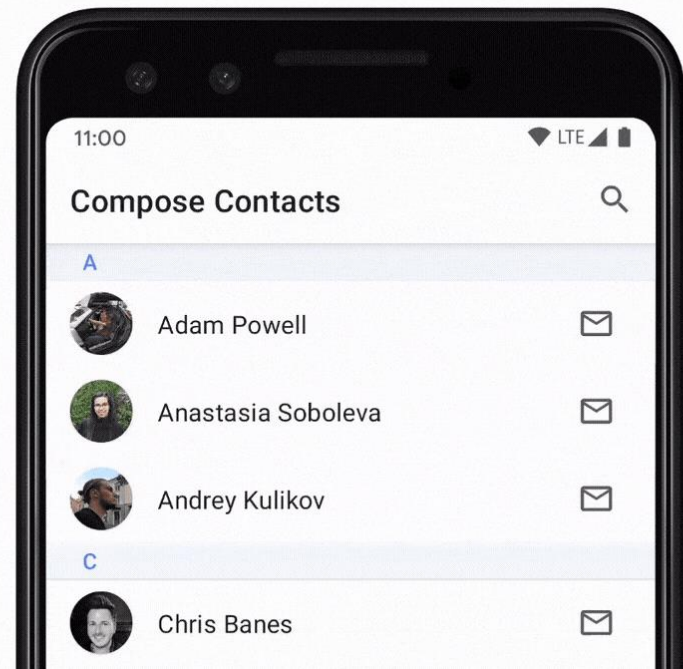
```
LazyVerticalStaggeredGrid(  
    columns = StaggeredGridCells.Adaptive(200.dp),  
    verticalItemSpacing = 4.dp,  
    horizontalArrangement = Arrangement.spacedBy(4.dp),  
    content = {  
        items(randomSizedPhotos) { photo ->  
            AsyncImage(  
                model = photo,  
                contentScale = ContentScale.Crop,  
                contentDescription = null,  
                modifier = Modifier.fillMaxWidth().wrapContentHeight()  
            )  
        },  
        modifier = Modifier.fillMaxSize()  
    )  
)
```



StickyHeader

```
@OptIn(ExperimentalFoundationApi::class)
@Composable
fun ListWithHeader(items: List<Item>) {
    LazyColumn {
        stickyHeader {
            Header()
        }

        items(items) { item ->
            ItemRow(item)
        }
    }
}
```



```
@OptIn(ExperimentalFoundationApi::class)
@Composable
fun PagerBasics() {
    val pagerState = rememberPagerState()
    HorizontalPager(
        pageCount = 3,
        state = pagerState,
        pageSize = PageSize.Fill
    ) { page ->
        Text(
            text = "Page: ${page + 1}",
            modifier = Modifier
                .fillMaxSize()
                .wrapContentSize(Alignment.Center), // centrowanie wertykalne
            textAlign = TextAlign.Center, // centrowanie horyzontalne
            fontSize = 36.sp
        )
    }
}
```

