

Java를 알고 C배우기

컴퓨터프로그래밍3

week 6-1 포인터와 배열-배열이름은 포인터다

2022.1학기
충남대 조은선

- (1) 배열이름과 포인터 관계
- (2) 배열 이름에 * 붙일 수 있다
- (3) 포인터에 [] 붙일 수 있다

배열 이름은 000다

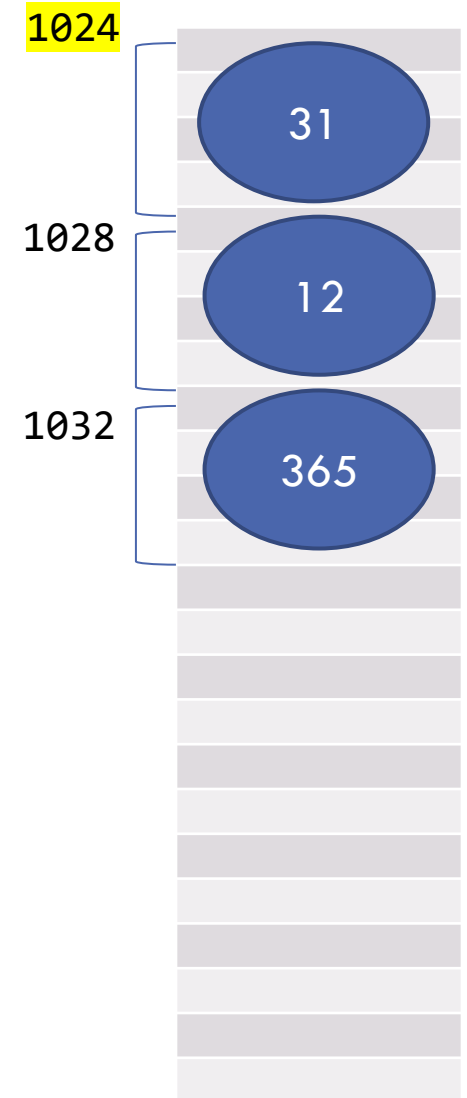
- ▶ 배열 이름은 배열의 시작주소 값을 의미하는 **포인터**임
- ▶ 배열 이름은 배열은 배열의 **첫번째 요소의 주소**로도 볼 수 있음 (예: **&arr[0]**)
- ▶ 대입연산은 불가능함 (변수가 아니라 상수임)

```
int main(){
    int arr[3] = {31, 12, 365};
    printf("배열 이름: %p \n", arr);
    printf("첫 번째 요소 주소: %p \n", &arr[0]);
    printf("두 번째 요소 주소: %p \n", &arr[1]);
    printf("세 번째 요소 주소: %p \n", &arr[2]);
    // arr = &arr[1]; // 오류!
    return 0;
}
```

같다

실행 결과

배열 이름:	1024
첫 번째 요소 주소:	1024
두 번째 요소 주소:	1028
세 번째 요소 주소:	1032



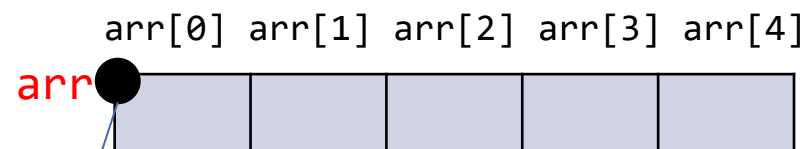
배열 이름을 포인터처럼 쓰기

- ▶ 배열 이름에 일반 포인터 처럼 * 연산을 해서 값을 가져올 수 있음
- ▶ arr이 배열이면, * arr 는 첫번째 요소 arr[0]과 동일하게 사용됨

```
int main(){
    int arr[3] = {0, 1, 2};
    arr[0]++;
    printf("%d \n", *arr);
    printf("%d \n", arr[0]);

    *arr += 100;
    printf("%d \n", arr[0]);
    return 0;
}
```

같다



시작 주소가 1024라면
`arr == &arr[0] == 1024`

// 실행결과:

```
1
1
101
```

포인터 변수를 배열처럼 쓰기

- ▶ 포인터에 배열의 시작주소를 지정해서 배열처럼 사용할 수도 있음

```
int main(){
    int arr[3] = {0, 1, 2};
    int * ptr = &arr[0]; // int * ptr = arr; 과 동일 — ptr과 arr은 같은 값을 가짐

    printf("%d %d\n", ptr[0], arr[0]); // 0 0 출력
    printf("%d %d\n", ptr[1], arr[1]); // 1 1 출력
    printf("%d %d\n", ptr[2], arr[2]); // 2 2 출력
    printf("%d %d\n", *ptr, *arr);      // 0 0 출력

    return 0;
}
```

포인터 변수인 ptr을
배열 이름 처럼 여기고
ptr[0] , ptr[1]... 등으로 사용 가능

Quiz

- ▶ 다음 프로그램이 출력하는 것은? (모르는 초기 값은 ?로 나타내시오.)

```
int a[] = {0, 1, 2, 3, 4};
int main() {
    int i, *p;

    for ( i=0; i<=4; i++)          printf("%d \n", a[i]);
    for ( p= &a[0]; p <= &a[4]; p++) printf("%d \n", *p);
    for ( p= &a[0], i=1; i<=5; i++)  printf("%d \n", p[i]);

    // 아래는 6-2의 Quiz까지 공부한 다음에 풀어보는 것이 좋습니다
    for ( p=a, i=0; p+i <= a+4; p++, i++) printf("%d \n", *(p+i));
    for ( p=a+4; p>=a; p++, p--)          printf("%d \n", *p);
    for ( p=a+4, i=0; i<=4; i++)          printf("%d \n", p[-i]);
    for ( p=a+4; p>=a; p--)               printf("%d \n", a[p-a]);
}
```

(참고)
연산자 우선순위
높은 순서
f(), [], ., ->, x++
&x, *p, ++x, -x
, /, %
+, -
<=, <, >, >= ...
==, !=
&&, ||
=, +=
,